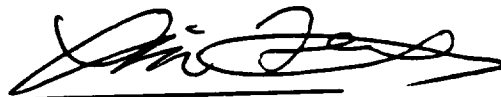A Final Technical Report on

Advanced Display and Computer Augmented Control System (ADCACS) for Space Station Freedom Body-Ported Cupola Workstation (BP/VCWS)

Submitted to: Dr. Michael W. McGreevy
                 NASA Ames Research Center (Grant NCC-2-681)

Prepared by:

Xin Feng, Ph.D.
Principal Investigator
Department of Electrical and Computer Engineering
Marquette University
Milwaukee, WI 53233
(414)288-3504

Marquette University
Research Team:
                 Dr. David Krubsack, Research Associate

                 Dr. Lee Belfore, Faculty Consultant
                 Dr. Ronald Brown, Faculty Consultant

                 Miss. Michelle Gebheim, Research Assistant
                 Mr. Roger Hermann, Research Assistant
                 Mr. Raghu Santamavattur, Research Assistant
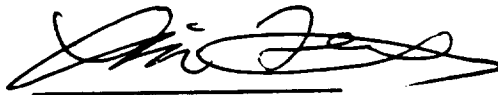   Mr. Kang Yin, Research Assistant

A Final Technical Report on

Advanced Display and Computer Augmented Control System (ADCACS) for Space Station Freedom Body-Ported Cupola Workstation (BP/VCWS)

Prepared by:

Xin Feng, Ph.D.
Principal Investigator
Department of Electrical and Computer Engineering
Marquette University
Milwaukee, WI 53233
(414)288-3504

Marquette University
Research Team:

Dr. David Krubsack, Research Associate

Dr. Lee Belfore, Faculty Consultant
Dr. Ronald Brown, Faculty Consultant

Miss. Michelle Gebheim, Research Assistant
Mr. Roger Hermann, Research Assistant
Mr. Raghu Santamavattur, Research Assistant
Mr. Kang Yin, Research Assistant

Table of Contents

CHAPTER THREE   **A COMPLEMENTARY REPORT ON COMPUTER ENGINEERING RESEARCH** 3-1

CHAPTER SIX STUDIES OF VIRTUAL DISPLAY GRAPHICS ENGINE ARCHITECTURE

CHAPTER SEVEN   HUMAN FACTORS RESEARCH AT ACT LABORATORY

CHAPTER EIGHT   LITERATURE SEARCH AND PARADOX DATABASE

List of Figures

ABBREVIATIONS

AAP	Advanced Automation Program
AAT	Advanced Automation Technology
ACA	Astronautics Corporation of America
ACT	Advanced Control Technology
ADCACS	Advanced Display and Computer Augmented System
BP/HPC	Body-Ported High Performance Controller
BP/VCWS	Body-Ported Virtual Workstation
C&WS	Caution and Warning System
CCTV	Closed Circuit Television
CCZ	Command and Control Zone
CERS	Crew/Equipment Retrieval System
CLIPS	C Language Integrated Production System
CRT	Cathode Ray Tube
CTS	Clear to Send
CWS	Cupola Workstation
DMA	Direct Memory Access
DMS	Data Management System
DSP	Digital Signal Processing
EDP	Embedded Data Processor
EIA	Electric Industries Association
EOAT	End-of-Arm-Tooling
ETS	Expertise Transfer System
EVA	Extra-Vehicular Activity
FBM	Force-Based Master
FBM	Frame Buffer Memory
FDDI	Fiber Optic Distributed Interface
FFE	Full Function Emulator
FOHMD	Fiber-Optic Helmet Mounted Display
GEA	Graphics Engine Architecture
GPC	Geometry Processor Component
HLGP	High Level Graphical Primitive
HMD	Head Mounted Display
IHADSS	Integrated Helmet and Display Sight System
ISO	International Standards Organization
IVA	Intra-Vehicular Activity
JSC	Johnson Space Center
KBS	Knowledge Based System

LAN	Local Area Network
LCD	Liquid Crystal Display
LCS	Liquid Crystal Shutter
LEEP	Large Expanse Extra Perspective
LLGP	Low Level Graphical Primitives
LSCE	LEEP Scan Conversion Engine
MFLOPS	Milliam Floating Point Operations Per Second
MIMD	Multiple Instruction Multiple Data
MIPS	Milliam Instructions Per Second
MPAC	Multi-Purpose Application Console
MSFC	Marshall Space Flight Center
MSU	Mass Storage Unit
MTBF	Mean Time Between Failure
NFS	Network File System
NIA	Network Interface Adapter
NIU	Network Interface Unit
NSTS	National Space Transportation System
NTSC	National Television Standards Committee
OOD	Object Oriented Design
OOP	Object Oriented Programming
ORU	Orbit Replaceable Unit
OSI	Open System Interconnect
PBM	Position Based Master
PPC	Pixel Processing Component
RISC	Reduced Instruction Set Computer
RMS	Remote Manipulator System
RPC	Remote Procedure Call
RTS	Request To Send
SCSI	Small Computer System Interface
SGI	Silicon Graphics Incorporated
SIMD	Single Instruction Multiple Data
SSF	Space Station Freedom
SSFP	Space Station Freedom Program
SSPE	Space Station Program Element
TGDS	Time Generation and Distribution System
VCASS	Visually Coupled Airborne Systems Simulator
VLSI	Very Large Scale Integrated Circuit
VMK	Voice Master Key
WFOV	Wide Field of View
ZMM	Zero Motion Master

CHAPTER ONE
GENERAL DESCRIPTIONS

1-1.0   Organization of the Report

The ADCACS Final Technical Report consists of nine chapters. Chapter One contains a general description of technical results and research efforts of the Marquette University research team, supported by the Astronautics Corporation of America. Technical details are enhanced and complimented by Chapters Two through Chapter Nine of the report.

Chapter One is organized to correspond to the Project Description, as outlined in the Section two of the original ADCACS project proposal. This will facilitate monitoring the progress of the proposed research.

Not all subsections in Chapter One correspond exactly to the original research proposal. Research results have modified several subsection listings.

1-2.0   Project Description

The subsection numbers in this section have been chosen to correspond to the section numbers in the original ADCACS project proposal. Consequently, the numbering scheme is not contiguous.

1-2.1.1 Overall Project Objectives

The Advanced Display and Computer Augmented Control System (ADCACS) was proposed as an advanced, three-year Space Station research project proposed by Marquette University (MU) and Astronautics Corporation of America (ACA), and sponsored by NASA Ames Research Center. Despite the fact that the objectives of the ADCACS project have been defined in Section two of the original proposal and have been carried out since the beginning of the project, the following section reflects a better understanding of these objectives.

The overall objectives of the ADCACS project have been to:

1) demonstrate the feasibility and practicality of the proposed Body-Ported Cupola Workstations (BP/VCWS) to meet the needs of the baseline and evolutionary Space Station Freedom (SSF) Cupola Workstations (CWS);

2) initiate an engineering approach to construct such a BP/VCWS, and through evolution, to replace existing "hard console" cupola workstation design;

3) develop a flight-qualifiable prototype BP/VCWS;

4) significantly advance the general state-of-the-art in workstation technology including "Virtual Reality" and "Intelligent Control" technologies.

1-2.1.2 Phase One Objectives

The Phase One research, started July 1, 1990 and ended June 30, 1991, has been carried out at the Marquette University Advanced Control Technology (ACT) Laboratory, with participation of Astronautics Corporation of America (ACA). The objectives of the Phase One research have been to:

1) define requirements, baseline and evolutionary configurations for the BP/VCWS that are consistent with crew user requirements and program constraints;

2) study the relevant technologies that will be applied to the proposed BP/VCWS, and to evaluate the feasibility of involved technology by conducting literature studies and laboratory investigations;

3) define design accommodations (Hooks and Scars) to satisfy evolutionary requirements for advanced human-system interfaces, and to specify body-ported virtual display system and intelligent control systems;

4) design the Full-Function Emulator (FFE) for further evolutionary studies of developing flight-qualifiable BP/VCWS.

Phase One final deliverable items have been specified as:

1) A Preliminary Prototype BP/VCWS Design;
2) The Full-Function Emulator System Design;

3) Preliminary and Final Design Accommodations (Hooks & Scars) Reports;
4) Final Technical Report.

Marquette University responsible items have been identified as:

1) The Final Technical Report that covers research results from activities 3, 6, 7, and 8, and part of activity 5 (see Figures 1-2 and 1-3 and Section 1-2.7.2 for details).

2) The BP/VCWS Full Function Emulator (FFE) System Design Report (Activity 11).

1-2.4   A Review of General Plan of Work

1-2.4.1   Systems Engineering Research

1-2.4.1.1   Summary of Systems Engineering Research

Following is the systems engineering work accomplished by the Marquette University sub-team during Phase One research:

1) Literature Search and Analysis

A literature search was carried out to learn relevant display, control, and other supporting technologies. Among hundreds of literature references that have been identified and used for the experimental designs of the ADCACS project, 85 most closely related titles have been entered into the PC-based PARADOX database. The listing of the titles are presently included in Chapter Eight of the report. Also a detailed virtual reality technology survey is included in Chapter Two, Section 2-1.0.

2) A Systematic Review of Existing Research Projects

Several existing and on-going research projects have been investigated. Among them the following projects are found to be closely related to the ADCACS and have been intensively studied:

a) VIVED workstation system and EVA Helmets (NASA/Ames/JSC);
b) Space Station Advanced Automation Program (NASA);
c) Integrated Helmet and Display Sight System (IHADSS, Honeywell);
d) Visually Coupled Airborne Systems Simulator (VCASS, Wright-Patterson Air Force Base);

e) Fiber-Optic Helmet Mounted Display (FOHMD, Williams Air Force Base);
f) "Super Cockpit" System (US Air Force);
g) "Pilots Associate" System (McDonnell Douglas and Lockheed);
h) "C" Language Integrated Production System (CLIPS, NASA/JSC).

3) A preliminary design accommodations (Hooks and Scars) report has been completed and submitted.

4) A detailed systems engineering study description, including a related technology survey report, a review of baseline CWS requirements, and an overview of the proposed BP/VCWS architecture is also included in Chapter Two of this report.

In addition, a preliminary requirements specification has also been completed and submitted (Section 2.7, Activity 2).

1-2.4.1.2 Conclusions and Recommendations

Based on the results from the ADCACS systems engineering research, the following conclusions can be made:

1) The application of virtual display and knowledge-based intelligent control technology to the Space Station Workstation design is a viable approach to increase the productivity and reliability of SSF cupola operations and enhance crew safety. The research and development activities for the construction of the proposed BP/VCWS that comprises start-of-the-art virtual display and intelligent control technologies are feasible.

2) Considerable risks, however, do exist. It is the perception and experience of the MU ADCACS research team that evolutionary development approach must be taken toward the construction of the proposed BP/VCWS, starting from building the on-ground emulator, due to technical, schedule, and cost risks. This is the only approach that makes the previously defined ADCACS project goals achievable, while reducing risks to an acceptable level.

3) Major risks can be specified as follows:

a) Display Devices

The desire for a more efficient and productive body-ported and virtual display device includes the requirement of a head mounted, Wide-Field-Of-View (WFOV) display device, with 1000 by 1000 pixels high resolution and

stereoscopic, color display. It is also required that the head-mounted device is light in weight and low in electric power consumption. Current existing display technologies does not meet the above requirements.

Among all technology candidates, the Distributed Cathode Display technology is considered as a promising technology that will meet the requirements of the body ported workstation. The risk in selecting this technology for display is that the technology is still in development, and substantial information is not available for testing and evaluation.

Another risk for selecting the Distributed Cathode Display is the development schedule of such a device. Though it is projected that the device will be available in the next few years, detailed schedule information was not available to the ADCACS project.

b) Knowledge Acquisition for the Knowledge-Based Intelligent Control

Knowledge Acquisition is referred to as the process of knowledge transformation from domain experts to the computerized knowledge base. This has been considered as a bottle neck for the expert system development. Cost risks do exist for using knowledge-based intelligent control, since the process of building the knowledge base in the expert system may be very expensive (see Chapter Five for details).

c) Computing Power for Graphics and Image Rendering

The major risk factor in this project is the lack of sufficient computational power. For the present time, realistic realtime images which are indistinguishable from what the human eye sees are just not possible. As an example, rendering images on a 386-based microcomputer, which are comprised of approximately 4 million polygons, would require approximately 60 minutes, per frame. At this rate, to produce a 60 second animation sequence, at a minimum of 24 frames per second, would require around 720 hours of rendering time alone.

Current virtual reality systems sacrificed detail for speed to allow the user to move through the environment with an interactive frame rate. The images resemble the appearance of "cartoons". VPL Research's RB2 system offers the ability render images at a rate of approximately 20 frames per second. Autodesk's virtual reality system operates approximately 10 frames per second. These frame rates are divided between two eyes. Even with these

relatively slow speeds, the capabilities of these systems are extraordinary.

It is only a matter of time though before the necessary computational power is available to handle more complex computational requirements. As an example, in February of 1989, Intel introduced their new 80860 (i860) chip. This VLSI processor is a 1 million transistor, 64 bit microprocessor which incorporates both integer and vector floating point units and 3-D graphics capabilities. Numerous vendors are incorporating this chip into their product lines.

Another area that has been receiving a considerable amount of attention recently is the use of massively parallel transputer based processing systems. Division Ltd. of Bristol U.K. has developed a virtual reality system based on Inmos transputers and Intel i860 microprocessors.

A detailed report on computational aspects for graphics and image rendering is included in Chapter Six of this report.

4) Cost Risks

Cost for developing the proposed BP/VCWS poses another challenge. Following examples illustrate the situation:

a) One of the most advanced virtual reality system in the world is located at **Wright-Patterson Air Force Base** in Ohio. This system utilizes a series of mainframe computers to create its virtual environments. Cost: Several million dollars (for the computers alone).

b) A midrange (although easily within the bounds of the "high-end" category) is the **Reality Built for Two** system (RB2) offered by **VPL Research**. Cost: approximately $225,000 for one user, about $430,000 for two. The primary expense of this system is in the graphics workstations. Two are required for each user (one for each eye).

c) The **Autodesk Cyberspace** system would be considered a low-end system. It could be assembled for approximately $40,000-$50,000. Images are generated by a 80386 computer outfitted with a pair of graphics accelerator cards as well as other assorted devices.

Following are general recommendations from the Principal Investigator:

1) ADCACS is a system engineering project that takes advantage of available state-of-the-art technologies, and performs system integration for the construction of the proposed BP/VCWS. It is not an independent, stand-alone project, and it is not built from the ground. Thus, the incorporation and applications of newly developed results and information from other closely related on-going research projects are critical and very important to the success of the ADCACS project.

   It is recommended, for this reason, that coordination between the ADCACS project and other related on-going NASA and government sponsored projects be enhanced.

2) Reduction of Major Risks

   a) The well developed Cathode Ray Tube (CRT) display is a mature and useful technology for color and high resolution. Because of availability, presentation capabilities, and ease of interface, ADCACS plans to use CRTs for all experimental and emulator designs, which is the first step of evolutionary development. (see Chapter Four for detailed display technology analysis). It is also recommended that the CRT display technology be considered as the backup technology for the body ported workstation.

   b) The automated knowledge acquisition tools are recommended for experimental testing and for use in the development of the Knowledge-Based System (KBS) to reduce the cost of software development. See Chapter Five for details.

   c) High power Intel i860 RISC microprocessors should be employed for the Graphics Engine Architecture (GEA, see Chapter Six for details). More advanced microprocessors should be employed during the BP/VCWS evolution.

1-2.4.2 Computer and Electronic Engineering Research

1-2.4.2.1 Major Work in the Marquette University ACT Laboratory

1) Several major hardware items have been installed for the development of a prototype emulator as the first step of the system evolution. Major equipment

includes a Silicon Graphics Super-personal IRIS workstation with stereoscopic display controller, a 33MHz 386 PC, and a 286 PC, networked by Ethernet.

2) A PC-based speech recognition system called Voice Master Key, manufactured by Convox, Inc., had been installed and tested successfully at the beginning of the ADCACS project. A 386 PC is designated for knowledge-based expert system control with the input from voice recognition system. The voice recognition accuracy seems exceedingly satisfactory. The immediately available voice synthesis is understandable, but not very high in quality. Parameters may be adjusted to improve the voice synthesis quality. Simple playback of recorded message is possible.

3) The NASA CLIPS and the commercial NEXPERT OBJECT expert system tools have been installed and tested for investigations of intelligent control. Knowledge-based expert systems appear to offer significant cost effective opportunities in terms of increasing the productivity of human operators of complex systems. In order to overcome the "knowledge acquisition bottleneck", a special methodology called "repertory grid" is suggested for knowledge acquisition and knowledge representation. "Multi-expert system" is seen as an advantage over single expert systems since decisions are based on the knowledge of a group of experts instead of simply a single expert.

4) A network-based prototype virtual workstation emulator for experimental studies of virtual display and knowledge-based intelligent control has been successfully developed (see Chapter Three for details).

5) A system design of the Full Function Emulator (FFE) has been completed for the construction of a Full Function Emulator for the proposed BP/VCWS (see the separate design report for details). It is planned that the research for developing the FFE be continued in Phase Two of the ADCACS project should the funding become available.

1-2.4.2.2   Research in Computing Technology for BP/VCWS Design

A detailed description of computing technology research is included in Chapter Three of the Report.

1-2.4.3 Display Technologies Research

1-2.4.3.1   Display Devices

1) The study of display requirements for the proposed BP/VCWS has been completed. Several existing display technologies, including Liquid Crystal Display (LCD), Cathode Ray Tube (CRT), Distributed Display, and Private Eye Display technologies, have been investigated. Following is a summary of display research:

a) Liquid Crystal Display Technology

In Liquid Crystal Display (LCD) technology, state-of-the-art resolution is limited to values well below that of a conventional color television or less than 500 lines. A second disadvantage of liquid crystal display systems is their low optical transmissivity which leads to the requirement for back illumination systems that require significant amounts of power.

b) Cathode Ray Tube Technology

Cathode Ray Tube (CRT) technology is a well developed technology for color and high resolution. The ADCACS project plans to use CRTs for all experimental and emulator designs because of availability, presentation capabilities, and ease of interface and use. The present state-of-the-art in small displays (1" diameter), is limited to monochrome due to the color mask. Color beam index CRT is a projected technology that will be useful for small displays.

c) Distributed Cathode Display Technology

Distributed Cathode Display Technology continues to be a promising technology for the body ported workstation. Since the research is still in the early stages, it has not yet been possible for the ADCACS project to obtain any substantial information.

d) The Private Eye Technology

The Private Eye Technology is not presently being pursued for several reasons. It requires a 2-3 minute training period for your eye and brain to get accustomed to the display. Presently, only monochrome (red) is available. Stereo could possibly be generat-

ed using two devices, but alignment would be difficult. The resolution is only 720 by 280. It has no WFOV capabilities. It has several moving parts which raises the question of reliability.

2) Conclusions:

a) The desire for a more efficient and productive body-ported and virtual display device includes a 1000 by 1000 pixel high resolution, stereoscopic, color display device that is light in weight and low in electricity consumption. The current existing display technology does not meet the above requirements for such a head mounted, Wide-Field-Of-View (WFOV) display device.

b) Among all technology candidates, the promising Distributed Cathode Display technology is considered as the priority technology that will meet the requirements of the body ported workstation. The risk for selecting this technology for display is that the technology is still in development, and substantial information is not available for testing and experiment.

1-2.4.3.2   Research on Optical Devices

The choice that the Marquette University ACT lab has made for the emulator design is the LEEP stereoscopic viewer optics that are produced by Eric Howlett of POP-OPTICS LABS. His viewing optics are made from six large glass elements which allow one to get a 90 degree direct field-of-view and a total field-of-view of 140 degrees when the corneal field is taken into account. This larger total field-of-view results from refraction at the surface of the cornea. This is a very important aspect of realistic immersion in the scene even though the peering field is no longer available.

Following are the summaries of optical system research which are detailed in Chapter Four:

1) The LEEP optics suffer from chromatic distortions due to its nonlinear transformation. These can be compensated for by varying the size of the different color images. The main problem with this is that it will add to the computational overhead of the system, slowing recalculation of the images.

2) The pre-warping required in order to obtain the desired high quality central image is another computational burden again slowing recalculation of images.

3) The LEEP format is not presently available as a standard option on the Personal IRIS. It may be possible to render the virtual elements into a LEEP format using

software and by-passing the graphics engine. This would require more processing time to render the picture and real-time manipulation would probably not be possible. Optic and human factor experiments could possibly be performed to analyze the LEEP viewing format.

Also there are concerns about the computational burden created by using LEEP optics and high density graphics generation and rendering. The greatest computational burden is the additional graphical computations necessary to restore the correct perspective and color to an image viewed through the LEEP optics. This problem could be solved by designing a dedicated hardware to perform the LEEP transformation. This would relieve the graphics engine of that task and speed recalculation of the image. A Complementary design report of the proposed LEEP transformation hardware experiment is included in Chapter Four, Section 4-4.

Research of using dedicated graphics engine, i.e., the Graphics Engine Architecture (GEA) with the ability to do faster image recalculation with the ability to perform scaling, has been performed. It is recommended that high performance Intel i860 RISC microprocessor and parallel computing technologies be considered for developing the GEA. The detailed studies, including parallel architecture and graphics generation and rendering algorithms, are included in Chapter Six.

1-2.4.3.3   Laboratory Investigations on Virtual Display and Computer Graphics

After analysis of graphics generating workstations, the Silicon Graphics (SGI) IRIS series of computers was found to provide the most cost effective graphical computational power. In this series, the Personal IRIS was within the ADCACS budget. A higher cost, more powerful VGX graphics generation system is available.

IRISVISION is an SGI product that provides a means whereby the graphical development on the Personal IRIS can be run on a PC. IRISVISION is a two-board set for a Microchannel or AT bus. The system uses the IRIS Graphical Library which is the library presently being used for the ADCACS project. IRISVISION began shipping in January 1991.

The projected WFOV setup does not require head tracking or non-linear transformation. It is projected that with the available time and resources left on the ADCACS project, non-stationary WFOV will not be obtained.

The projected stationary WFOV requires several optical elements including mirrors and cylindrical, warping, and WFOV lenses. For body ported, non-stationary WFOV, many optical elements need to be removed. With appropriate display technologies, the

mirrors and cylindrical lenses could be removed. DVI technology can perform warping after rendering. A technology is needed to perform pre-warping in real time using computer hardware.

The graphics programs have advanced to a state where human factor considerations can be investigated in an operational mode. The response time between control input and graphical generation has decreased to a point where there is no perceivable delay. More information of the input devices can be found in the following sections. This section is devoted to current graphical generation techniques.

Graphical generation is performed on a Silicon Graphics Personal IRIS 4D/35TG (33 MHz CPU). This machine is fast enough to generate real-time (no perceivable delay) images necessary for the present research. It is anticipated that as the complexity of the control scenarios increases, the graphical image generation power will have to increase.

The primary graphics program in use at the present time is "armfgrbox.c". It uses the Silicon Graphics standard graphics library, GL. GL is a set of C callable subroutines. The subset of routines being used in "armfgrbox.c" includes: 1) a variety of viewing windows; 2) a single infinite light source (multiple infinite and local lighting source models are available for experimentation and require increased processing time); 3) RGB mode and Gouraud shading for realistic shading with fewer polygons as compared with flat shading; 4) double buffering to synchronize rendering with the screen so smooth motion exists with no flickering; 5) backface removal and z-buffering to decrease rendering time for hidden surface removal; 6) viewing matrix manipulation and stack routines to easily and quickly position graphical elements; 7) queues for internal events and standard input devices; and 8) simple 3 dimensional polygon rendering routines.

"armfgrbox.c", as the name might imply, renders a picture of a right arm including a hand and fingers using positioning data received over the network. It also has techniques for drawing boxes in any location and orientation. For iteration experimentation of the graphics portion of the emulator, a second process may be run on the Personal IRIS which emulates a microcomputer sending data to the IRIS.

For the networking setup, the IRIS is the client and the fingers/hand/arm positioning preprocessor microcomputer is the server. The main graphic loop requests positioning data from the server. It then immediately renders a picture using initial data. After rendering the picture, it reads the data sent from the server and loops. The second time through the loop it again requests data from the server and immediately renders a picture, this time with the previous data. Each time through the loop the picture is rendered using the previous data. This technique allows the graphic generation to be performed in parallel with the acquisition and preprocessing of the positioning data (See

Figure 1-1).

The screen is refreshed at 60 Hz so there is no perceivable flicker. The image update time is approximately 8 Hz. Until the next image is generated, the IRIS simply redisplays the same image. For the same 3 dimensional model, the image generation time will differ depending on the depicted position of the virtual arm.

– Collision detection has been successfully completed. In addition to drawing an arm, several boxes are also drawn within arm's reach. The location of the tip of each finger is compared to the location of each box, resulting in a Boolean array of collision locations. Additional points on the fingers, hand, or arm may be identified as important for collision detection and added to the Boolean array. This may be done for any number of boxes.

Once the collision detection data is acquired, decisions may be based on the data. These decisions could include coloring the appropriate box, limiting motion of the moving object (a finger presently), or grabbing and moving one of the boxes. If these decisions are simple and need to be made quickly, they could be performed internal to the IRIS. If they are complex and are made with a longer time delay, it may be appropriate to off-load the decision process to another processor on the network to preserve processing time on the IRIS for complex graphical generation. The decision processor could be running a knowledge-based software package.

The present collision information is used to draw the appropriate box with a color that is dependent on which finger is touching the box. This gives the operator clear visual feedback.

A virtual model of a space cupola environment is being developed. An additional program has been written which allows the virtual arm to push boxes in the virtual space.

Collision detection was first attempted using the routines provided by the GL library. After some experimentation and communication with Silicon Graphics, a bug was found in the geometry engine's microcode on our specific machine. The bug was labeled "Bug number 11700". Our present collision detection technique avoids the bug and also allows the collision detection to be performed faster. The disadvantage is that a finger (exclusive of the tip) may be in contact with a box and not be detected. At the present state of the research, this is not a problem.

Additional detailed work on computer graphics experiments are included in Chapter Four.

1-2.4.3.4   Computational Aspects of Graphics Rendering Technology

The general display and graphics rendering technologies research for the proposed BP/VCWS are also included in Chapter Four.

1-2.4.4 Human Engineering Research

The Department of Defense and NASA Human Interface Conditions have been analyzed, and where appropriate, applied to problems in the prototype emulator. The human factor analysis has been separated into:

1)   input devices;
2)   display presentation, whether graphical or remote monitoring;
3)   system solution concepts.

Chapter Seven contains an analysis of human factors that is useful in the design and analysis of the experimental setups.

1-2.4.5 Research in Work Station Operator Interaction with Displayed Information for System Control

1)   Keyboards

Three different keyboards for data entry in the workstation were investigated. The first, the DVORAK keyboard, was a redesign of the standard keyboard. It basically reorganized the QWERTY keyboard so that the most often used letters were easier to reach and closer together. It was proven that once a keyboard was learned, the speed could be increased (by 5 to 10%). The drawback of this unit was that it required each finger to press multiple keys. This would not achieve the goal for this project where the most efficient way of data entry is desired.

The next keyboards that were investigated were the chording type keyboards. These keyboards have only 8 or 10 keys and by pressing these keys singularly or in combination the different letters are typed. The first chording keyboard investigated was the Wlonk keyboard; it has 10 keys which are previously defined for the user and the definitions must be learned. This keyboard had a programmable feature that would allow the user to enter frequently used macros into the keyboard's define system.

The other chording keyboard that was researched was the ternary chording keyboard. This keyboard has only 8 keys but each key has three positions. The qualities of this keyboard that were remarkable are the ease with which a system like this could be integrated into the Force-Based Master (FBM, see Chapter Nine for detailed information) hand assemblies. The hands are already located at a revised keyboard, the fingers never have to move from the eight keys for data entry, and if the mode of the FBM could be switched to data entry from anthropomorphic manipulator control, the input from the fingers could be passed through the circuitry in this keyboard and then typed to the screen. The second aspect that makes this keyboard more suitable is that the chords can be learned in a matter of hours (approximately 3 hours as stated by Vatell). Though this doesn't mean the capability for greater speed exists, the logical result is that a larger percentage of the training time can be spent on speed after all of the chords are learned.

2)   Typical "Hard Console" Input Devices

In micro-gravity, it is possible for the precision of mouse, trackball, spaceball, and joystick input systems to degrade. The precision associated with mounting these devices in a body ported way needs to be considered.

3)   Master Technologies

   a) Rigid Exoskeleton Technology - Position Replication Based

   This technology involves Astronautics proprietary information. Please see Chapter Nine for details.

   b) Soft Exoskeleton Technologies - Position Replication Based

   The cost of the VPL Datasuit and Dataglove technology precluded its availability to the ADCACS project. A derivative of the Dataglove, the Matell Power Glove, has been evaluated and three difficulties were identified. First, the operator's arm needs to be elevated so that the hand is within the range of the sensors. This causes fatigue after only several minutes. Second, the operator needs to reset the glove every once in a while to maintain maximum accuracy. Third, its best accuracy was judged to be inadequate to operate the prototype problem FTS.

c) Force-Based Master (FBM) Technology

This technology involves Astronautics proprietary information. Please see Chapter Nine for details.

d) Micro-motion Master Technology

This technology limits the operator's range of motion whereby a small motion is used to control a larger motion. No specific device evaluation and laboratory experiments have been performed.

e) Optical Master Technology

Optical master devices provide high accuracy, but are usually expensive. One such device is the head tracking device found in the FOHMD system.

f) Combinations

At this time, no conclusions have been made as to which master technology is better than others for the proposed BP/VCWS. It is suggested that combinations of a) through e) may be appropriate for the Full Function Emulator (FFE) design for the proposed BP/VCWS. While one technology may be appropriate for one appendage, another technology may be appropriate for another. Additional experimental study needs to be done for the master technologies to be incorporated into the BP/VCWS design.

1-2.4.6 Intelligent Control Technologies Research

The following aspects of the Intelligent Control research have been performed in ADCACS Project:

1) Feasibility study of SSF Advanced Automation Program for the applications of BP/VCWS.

2) Initial experimental investigations of Knowledge-Based System (KBS) development tools (NEXPERT Object and NASA CLIPS).

3) Analysis and Identification of intelligent control functions for the BP/VCWS.

4) Study of knowledge acquisition methodology for knowledge base construction.

A detailed report on intelligent control and knowledge-based system are included in Chapter Five.

1-2.4.7   Data Transmission Technologies Research

1) The present prototype emulator environment consists of 2 IBM compatible PCs (one 286 PC and one 386 PC), and a Silicon Graphics Personal IRIS, networked together using Ethernet. The 286 PC is used for acquiring and processing data from the fingers/hand/arm position sensing input devices. The Personal IRIS is used for the graphic generation of the virtual scenario elements to be manipulated. This data communication process has been established and is being evolved to improve operator control of the virtual elements. The 386 PC is designated as the processing base for the expert system. It also is be used as a processing driver for the speech recognition.

2) The Ethernet Link was chosen for the networking of the prototype emulator. For inter-process communication over the Ethernet, "Sockets" was chosen as the protocol for the prototype emulator. RPC (Remote Procedure Call) and NFS (Network File System) were also evaluated as options to Sockets.

A detailed report on the study of computer networking for the laboratory virtual workstation prototype emulator is included in Chapter Three.

1-2.5   Broad Design of Experiments Undertaken

A prototype emulator for the proposed BP/VCWS has been successfully developed as a result of the ADCACS laboratory experimental investigation. This prototype emulator environment consists of an 80386 personal computer (PC1), a 286 personal computer (PC2), and a Silicon Graphics Personal IRIS workstation (IRIS). Also, a Force-Based Master (FBM) arm and a position-based master hand (see Chapter Nine for details), a chording keyboard, and a voice recognition device have been integrated into the prototype emulator. The force- and position-based arm/hand master, the chording keyboard, and the voice recognition system have been interfaced to the PCs. A virtual "graphic arm" and a virtual "graphic hand" have been developed on the IRIS workstation. An Ethernet network has been established to link each of the computers together into a single functioning unit. Please see Chapter Three for detailed descriptions of the prototype virtual workstation emulator environment.

1-2.6   Experimental Methods and Procedures

Experimental methods and procedures take applicable portions of preliminary specifications. They are also used to measure relative success of the prototype emulator in meeting these requirements during the performance of representative prototype problems.

1-2.7   Work Plan

1-2.7.1   Responsibilities for the Proposed Activities

The entire research has been divided into 12 different activities that were pursued concurrently by Marquette University and Astronautics research sub-teams (Figure 1-2 and Figure 1-3). Following is the descriptions of activity items and the designated responsible parties.

1)   Marquette University Responsible Items:

    a)   Activity 3 (Literature Search)
    b)   Activity 6 (Experiment Program Design)
    c)   Activity 7 (Experiment Program Installation)
    d)   Activity 8 (Experimental Investigation)
    e)   Activity 11 (Full Function Emulator Design)
    f)   Activity 13 (Final Technical Report)

2)   Astronautics Responsible Items:

    a)   Activities 1 and 2 (Review and Specifications of BP/VCWS)
    b)   Activity 4 (Preliminary Design Accommodations (H&S) Report)
    c)   Activity 5 (Human Factors and Human Engineering)
    d)   Activities 9 and 10 (Preliminary BP/VCWS Design)
    e)   Activity 12 (Commercialization)

1-2.7.2   Description of Activities

1-2.7.2.1   Activities Performed at Marquette University ACT Laboratory

1)   Activity 3 -   Display and Control Technologies and Supporting Technologies Literature Search and Analysis

A literature search has been done to learn the capabilities, features and availability of display and control technology. Among hundreds of literature references that have been identified and used in the experimental designs of the ADCACS project, the 85 most closely relevant titles have entered into the PC-based PARADOX data base. The listing of the titles are presently included in Chapter Eight, which shows the type of information recorded for each entry. A word search may be performed on any field (Subject, Key Words, Author, Title, ..., Abstract/Citation).

2) Activity 5 -   CWS Human Factors/Human Engineering Requirements
                       Literature Search and Analysis

This activity was originally identified as Astronautic's responsibility. Some work, however, has been performed at Marquette University ACT Laboratory.

The human factors and human engineering databases have been built and several important references have been uncovered (e.g. Design Handbook for Imagery Interpretation Equipment, by Richard J. Farrell and John M. Booth). The Part 2 Design Specification and the Workstation Conceptual Design approaches which are to be developed provide the most effective support of the human operator in terms of:

a) the input information forms presented;
b) the range of control action forms;
c) the basis of the design of the workstation.

Progress has been made in these areas and subsequently incorporated in the design of the prototype emulator. More information can be found in Chapter Seven.

3) Activities 6 & 7 -   Experimental Program Design and Experimental
                        System Installation

The design and installation of the experimental investigations are primarily motivated by the prototype problem of the Flight Tele-Robot Server (FTS). Each of the components in the prototype emulator is connected together by an ethernet network. The expert system tools for building the intelligent control decision support system have been analyzed.

The Silicon Graphics Personal IRIS graphics workstation has been chosen, purchased, and installed as the major computing device in the ACT Laboratory. The Personal IRIS was found to have higher resolution and faster computational speed, as compared to other graphical workstations and PC based graphics generation

systems presently available and within the budget.

This activity was accomplished on the designated schedule. A detailed description of Activities 6 and 7 are included in Chapter Three.

Following is the listing of ACT Laboratory major equipment installations:

    I) Equipment Acquired on the NASA Grant:

        a) A Silicon Graphics Super-Personal IRIS 4D/35TG (33 MHz CPU) Workstation with 3-Dimensional, Stereoscopic Display;

        b) A 33 MHz 386 Personal Computer with Super-VGA Color Monitor;

        c) A 16 MHz 386SX Lap-Top Personal Computer;

        d) A HP-III Laserjet Printer;

        e) A Wide Angle LEEP Optical Viewing System;

        f) A Voice Recognition and Input Command Controller;

        g) The ACCUKEY Special Keyboard;

        h) A Mita Copier;

        i) Software.

    II) Equipment Provided by Marquette University:

        a) A 12 MHz 286 Personal Computer with printer;

        b) Software (DOS, Word Processing, Nexpert Object Tool, etc.).

    III) Equipment Provided by Astronautics Corporation of America:

        See Chapter Nine for details.

4) Activity 8 - Selected Advanced Control Technologies (ACT) Laboratory Experimental Investigations

The major achievements of experimental investigations have been the integration and implementation of the prototype emulator. See Chapter Three for details.

1-2.7.2.2   Astronautics Activities

Research activities which were the responsibility of Astronautics will be reported separately and will be included in Chapter Nine.

1-2.7.3 Team Work at Marquette University ACT Laboratory

1) Figure 1-4 shows the Marquette University ADCACS research team organization.

2) Time Allocation of Research Sub-topics:

    a) 15%   BP/VCWS Systems Engineering Research.
    b) 15%   Intelligent and Knowledge-Based Systems Control.
    c) 70%   Virtual Display and Optics;
                  Master Command/Input Devices;
                  Human Factors;
                  Voice Recognition.

Serial Programing

client

server

IRIS PC IRIS PC IRIS

t_r

Concurrent Programing

client

server

IRIS PC IRIS PC IRIS PC IRIS PC IRIS PC IRIS PC

t_r
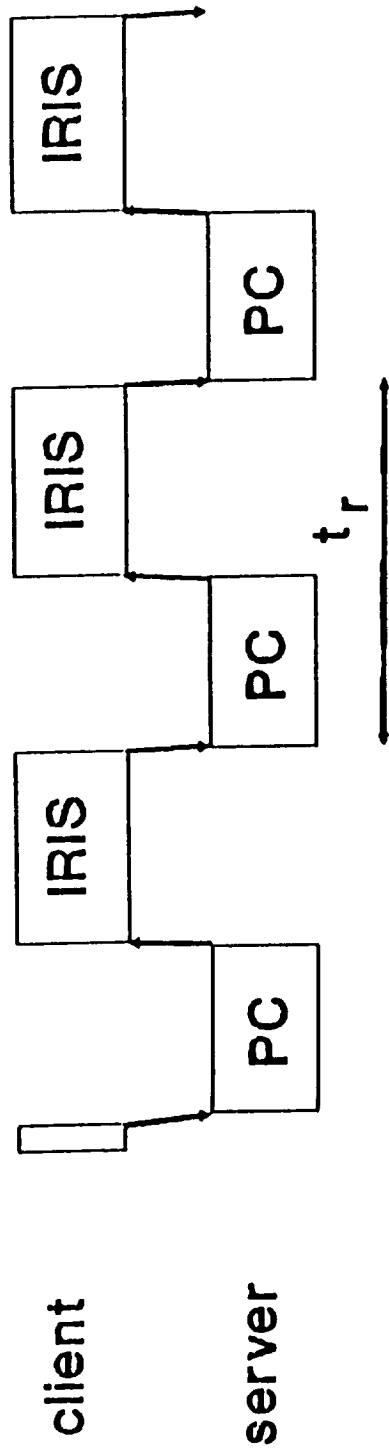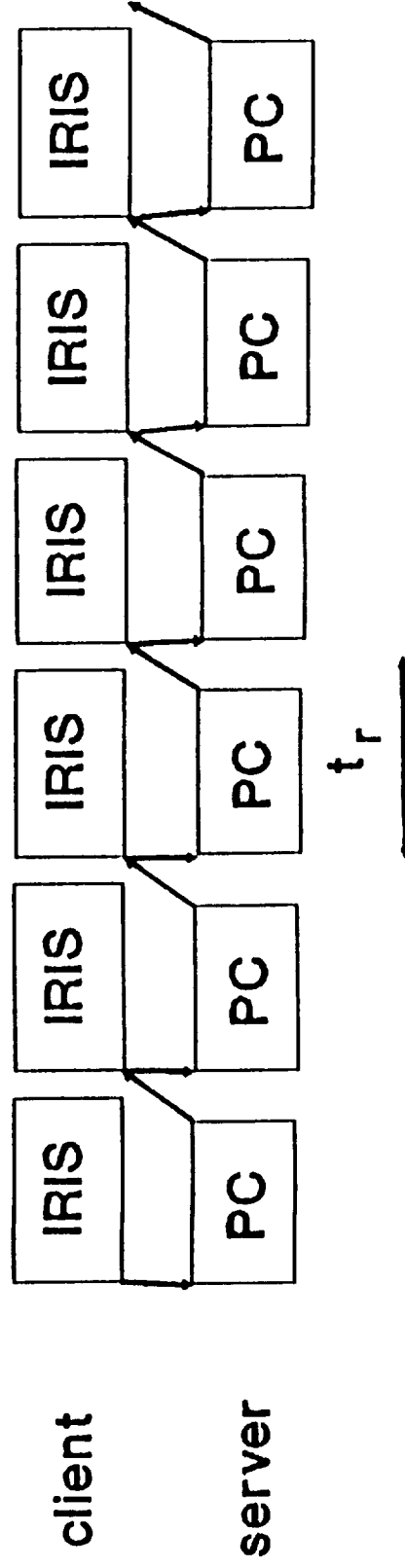
t_r = time to render a picture

Figure 1-1 Serial versus Parallel Processing of Network
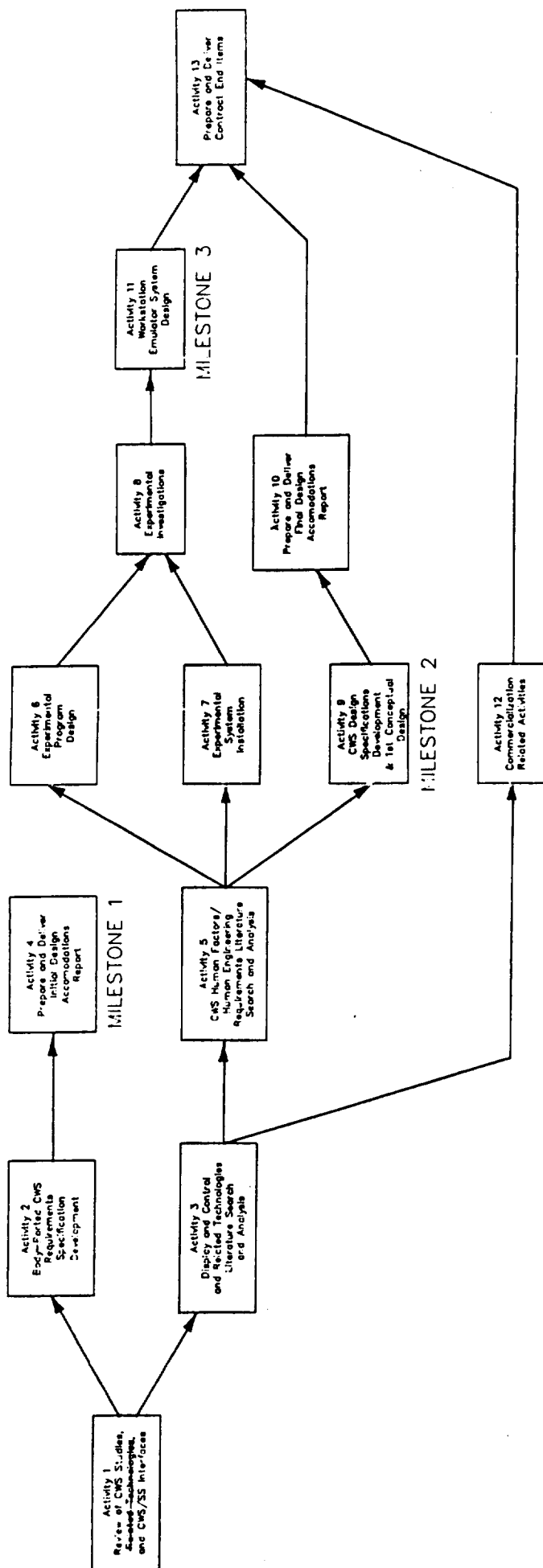
Figure 1-2 ADCACS Project Work Plan

The proposed project work flow and
project milestones in PERT format

Act #1 – CWS Studies, Related Tech, CWS/SS Interfaces Anal

Act #2 – Body-Ported CWS Reqts Spec

Act #3 – C&C and Related Tech LH Search/Anal

Act #4 – Deliver 1st Accomodations Rpt

Act #5 – Human Factors/Engineering Reqts And

Act #6 – Experimental Program Design

Act #7 – Experimental System Install

Act #8 – Experimental Investigations

Act #9 – CWS Design Spec Development and Conceptual Design

Act #C – Final Design Accomodations Rpt

Act #11 – Emulator System Design

Act #12 – Commercialization Activities

Act #3 – Deliver CEI's

MONTHS ELAPSED FROM PROJECT START

1	2	3	4	5	6	7	8	9	10	11	12
JUL	AUG	SEPT	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN

Figure 1-3 ADCACS Project Work Schedule

Marquette University
Department of Electrical and Computer Engineering

ADCACS Technical Team Organization
October 11, 1990

Dr. Niederjohn
ACT Lab Director

Project Coordination
Lab Equipment Management

Dr. Feng
Principal Investigator

General Management
Systems Engineering
Software Engineering
Intelligent Control

Dr. Krubsack
Research Scientist

Virtual Display
and
Experimental Design

Mr. Yin
Research Assistant

System Engineering
Knowledge-Based
Display Control

Mr. Hermann
Research Assistant

Optical Display
and
Sensor Technology

Mr. Santamavattur
Research Assistant

Man-Machine Inter-
action, I/O, and
Networking

Miss Gebheim
Research Assistant

Human Factors
and
Human Engineering

Figure 1-4 Marquette University Research Sub-team Organization

CHAPTER TWO

A COMPLEMENTARY REPORT ON SYSTEMS ENGINEERING RESEARCH FOR THE PROPOSED BP/VCWS

2-1.0   A Survey of Virtual Reality Technology

2-1.1   Introduction

This section provides a comprehensive survey on the technologies currently integrated into various virtual reality systems. Although there are a fair number of sites conducting investigations into these different areas, there are very few commercial products currently available.

2-1.2   Head-Mounted Display Systems

Ivan Sutherland is credited with the development of the first head-mounted display (HMD) back in the 1960's. This system utilized a pair of miniature cathode ray tubes to display the stereoscopic images in a fashion which also allowed the user to still see his/her real surroundings. Special computer hardware was developed to generate the images which were in turn presented to the user. A direct mechanical linkage from the HMD to the ceiling, as well as a sonic positioning system allowed the users head position and orientation to be tracked. The system displayed wireframe images.

Presently, LCD's are the primary display devices in virtual imaging systems. One such system which displays high quality images is in use by researcher's at **NASA Ames Research Center**. Produced by **Citizens Watch Company**, the LCD's are twisted pneumatic, monochromatic, with a contrast ratio of 7:1, a 16 level grayscale, and passive pixels.

The array for each eye measures 2.6 inches horizontal x 2.0 inches vertical. The pixels are diamond shaped with a resolution of 320 x 240. Diamond shaped pixels provide spatial filtering of high frequency pixel edges. This results in edges which are much less distracting than arrays with similar resolution but rectangular pixels.

The optics used in this display is the **LEEP** wide angle optical system developed by **Eric Howlett** of **POP Optix Labs**. These optics allow the images displayed on a 2.7 inch

diagonal screens to cover an extremely large portion of the wearer's field of view: 120°
horizontally and vertically with a common binocular field of 90°. Predistortion of the
images is required to compensate for the effect of the lenses. The HMD uses a magnetic
positioning system produced by **Polhemus Inc.** of Colchester, VT.

VPL Research of Redwood City, California produces a commercially available head-
mounted display system known as the **EyePhone**. This system contains two separate
high resolution color LCD displays produced by Sony. The EyePhone also utilizes the
LEEP wide angle optical system, providing a wide viewing field (100° for both eyes).

It has been indicated that VPL Research will be releasing a new model of the
EyePhone system in the very near future. This new display device is expected to be of
a more comfortable design, as well as provide for the use of headphones for 3-D acoustic
displays.

At Wright Patterson Air Force Base in Ohio, researchers have developed a head-
mounted display system referred to as the "SuperCockpit". Aircraft system information
is sent from a digital bus to a special digital signal processor (DSP) and graphics
generator. A special software package referred to as mindware translates these signals
into special sight, sound and touch patterns.

Visual patterns are displayed on miniature cathode ray tubes mounted inside of the
flight helmet. The image is directed into a set of optical components where it is
magnified, collimated and projected directly onto the pilot's eye. The pilot is able to see
through the images to his normal world. Daytime system use allows the pilots normal
view to be enhanced with additional information. Night or limited visibility use allows the
pilot to experience a total computer generated representation of the outside world.

A highly advanced head-mounted display project beginning at the University of
Washington in Seattle utilizes methods similar to those found in the SuperCockpit system.
See page 33 for a description.

Other sites conducting investigations in the area of head-mounted displays for
simulation include:

* **Rediffusion Simulation Ltd., United Kingdom**
* **CAE Electronics, Canada**
* **NASA Johnson Space Center, Houston, Texas**
* **University of North Carolina, Chapel Hill, North Carolina**
* **Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, Ohio**

* **3-D Imagetek Corporation, Glendale, California**
* **Honeywell Military Aviation Systems**

2-1.3   Software/Mindware

A software system referred to as a virtual reality "**toolkit**" is under development at a number of research sites. This system is intended to allow the rapid prototyping of virtual environments, as well as objects within those environments. A few systems also call for the ability to design a virtual environment while within a virtual environmental.

While some of these research site's recent efforts have been concentrated on techniques involving the speed and detail of image renderings, such as the **University of North Carolina**, others, such as the **University of Washington** are emphasizing the development of a class of software referred to as "mindware", which provides functionality and compatibility with the user's natural cognitive abilities. Both are extremely important areas.

Of all of the institutions that are conducting research into virtual realities, two which are currently receiving a considerable amount of attention in the area of software are **Autodesk Inc.** of Sausalito, California and **VPL Research** of Redwood City, California.

As Autodesk is a leading supplier of computer-aided design software, there is a natural lead in this area. Their product **AutoCAD** is customized with **AutoLISP** to construct the three-dimensional virtual environment. Another package known as **AutoSHADE** is used to render the images as the user moves about.

VPL Research has developed several software packages for use with their virtual reality system known as **RB2 (Reality Built for Two)**:

RB2 Swivel

This software is a special version of the top selling solid modeler for the MacIntosh, **Swivel3D**. It is used for designing and placing objects in a three-dimensional virtual environment. This is the only package which allows the ability to link objects. These objects can be constrained in any position and orientation, and assigned ranges for each degree of freedom. This allows features such as ball joints, hinges and sliding parts. Combinations of smaller elementary objects can be used to model more complex objects.

Body Electric

This is a realtime animation package. Files created by RB2 Swivel are imported as 3-D wireframe models which can then be edited and merged so they form complex scenes. Body Electric defines the connections between the inputs such as the DataGlove and Polhemus magnetic positioning devices and the model of the virtual environment.

– These connections are established by selecting boxes representing functions contained in the library and creating links between inputs and outputs with arrows in a dataflow fashion. This allows the user to define the behavioral characteristics of objects in the virtual environment.

ISAAC

This is a realtime rendering package installed on the **Silicon Graphics IRIS** workstations used in the RB2 system. It provides Gouraud shaded polygon images of the virtual environments. Utilities also allow the scaling of the virtual environment, definition of various viewing perspectives and the ability to adjust the intraocular distances for the EyePhones.

Sense8 Corporation of Sausalito, California has developed a software package for the creation of virtual environments which will be capable of running on a variety of different platforms.

2-1.4   Position/Orientation Sensing

Virtual reality systems require position and orientation tracking of the users head. This is in order to properly synchronize the images displayed in the viewing system with the direction the users head is facing. There are numerous methods available for tracking objects. These include acoustical, mechanical links or lateral-effect photodiode. The method most often employed in virtual reality is magnetic positioning due to it's degree of accuracy, the ability to track position, as well as orientation, and it's operational area.

There is commercially available magnetic positioning system from **Polhemus Inc.** (Colchester VT.) which seems to be the system of choice. Known as the **3Space Isotrak**, the system consists of a pair of sensors, one moving and one stationary. Electrical current is pulsed through three orthogonally mounted coils which are housed in the source. This produces a magnetic field which is monitored by the sensor which is mounted on a manual interface of head-mounted display. The system allows the position (x, y and z) and orientation (roll, pitch and yaw) of the moving sensor to be tracked in

three-dimensional space.

The **3Space** system provides a relatively high degree of accuracy in measurement, but has a limited range of operation (up to 30 inches) before accuracy begins to decrease. The presence of large metallic objects (desks, cabinets) can strongly affect the performance of the system as well. It is common practice to map out the position of ferrous objects within the operational hemisphere of the system and correct for these interferences in the software.

Other sites conducting research in this area include:

* **Ascension Technology Corporation, Burlington, Vermont**
* **Telepresence Corporation, Vancouver, British Columbia CANADA**
* **University of North Carolina at Chapel Hill**

2-1.5   Tactile Feedback

As virtual realities are synthetic models of physical entities, it is considered essential, especially in teleoperated control systems, to allow the operator to experience the surface textures, bulk properties and dynamics of these models. the presence of such haptic cues augments the ability to create convincingly realistic computer generated environments.

There has been a considerable amount of research conducted in this area. Two of these methods appear to be receiving a significant amount of attention. one involves the use of metals referred to as "shape memory alloys". These metals, such as a titanium-nickel alloy, change their shape with increases in temperature. Introducing temperature changes in the metal is usually accomplished through electrical current.

It is indicated that arrays of small rods could be oriented to push against the fingertips when they are heated. One company which is conducting research in the use of memory metals for tactile stimulation is **TiNi Alloy Company** in Oakland, California.

The second technique involves the use of piezoelectric crystals which are sewn into the finger tips of a glove, as well as arrays on various portions of a close fitting body suit. These crystals vibrate rapidly when pulsed with electrical current.

Research in the area of force display is also underway at a number of sites around the world. Most of these projects involve the use of motor driven joysticks as the manipulator. A number of systems have been developed by the **Massachusetts Institute**

of Technology Media Laboratory. Tests conducted in the perception of texture, roughness, torque and applied endpoint force have produced promising results.

Other sites conducting research in this area include:

* University of North Carolina at Chapel Hill
* Cybernet Systems Corp., Ann Arbor, Michigan
* Israel Institute of Technology, Haifa, Israel
* University of Utah
* Institute for Sensory Research, Syracruse University
* Institute of Perception Research, Eindhoven, The Netherlands

2-1.6   Eye Tracking

A significant amount of research has been conducted by and for the U.S. Air Force in the area of eye tracking. The resulting technologies developed through these investigations have enabled a wide range of applications to be realized, both inside and outside of the military.

LC Technologies, Inc. of Fairfax, Virginia has developed an extraordinary application for this technology. The Eyegaze Computer System uses a video camera mounted below a control monitor to observe the position of one of the user's eyes. A low powered infrared LED mounted in the middle o the camera lens illuminates the user's eye and provides a bright spot where it reflects off the cornea. Based on the image of the eye's pupil and the location of the bright spot, sophisticated image processing software continually calculates where the user is looking on the computer screen.

The computer calculates the time that the user's eye dwells on a particular menu selection or button displayed on the monitor. This is referred to as "gaze duration". If the gaze duration meets the preprogrammed time, the computer will activate the corresponding command.

Iscan Inc. of Cambridge, Massachusetts has developed a corneal refection eye tracking system that can be mounted into helmet, as well as a fixed sighting system. An infrared light is reflected off of a ciochroic mirror (which is transparent to the user) and onto the cornea of the user's eye.

Sentient Systems Technology of Pittsburgh, Pennsylvania has developed a system known as the EyeTyper method in which commands are entered by focusing the eyes on LED's which are positioned like a checkerboard around the edges of a display

2-7

hanging in front of the user. A camera located in the center of the display measures the reflection of an infrared light on the cornea relative to a target worn on the face just below the eye.

2-1.7   3-D Sound Systems

–   The use of three dimensional sound cuing in virtual reality is of extreme importance. This is particularly useful in demanding information environments and high workload situations where visual cues are limited. Visual and auditory icons can be combined to present a greater sense of presence or realism in a fashion neither could accomplish alone.

What is considered to be the most advanced three-dimensional sound systems in existence has just recently been released by **Gehrig Research** of Toronto, Ontario, Canada. Known as the **FOCAL POINT** audio system, through the use of a normal pair of headphones, this system can generate an extremely compelling sounds with directional properties.

Using a **Polhemus** magnetic positioning system, the user's head position and orientation is tracked by the computer. This information is used by the system to stabilize the apparent source of the sound in three-dimensional space while the user move his/her head, just like the real world. The binaural sound technology is based on head-related transfer functions. This system utilizes a widely available accelerator for the Maxli as a signal processor, which provides a sampling rate of 44 KHz. Analog I/O is 16 bits, with internal processing at 24 bits, providing a computational accuracy of 144 dB.

This system is in use in a number of research projects relating to virtual environments and future aircraft cockpits.

Additional sites conducting research in binaural sound include:

* **NASA Ames Research Center in Mountain View, California.**
* **Wright-Patterson Air Force Base, Dayton, Ohio**
* **Crystal River Engineering, Groveland, California**
* **Ruhr University in Bochum, West Germany**
* **Rank Xerox EuroPARC, Cambridge, United Kingdom**
* **Xerox Palo Alto Research Center, Palo, Alto, California**

2-1.8   Command Input Devices

There are several new devices available, or under development, that enable more natural and intuitive methods of interacting with computers and complex as well as the information they supply.

VPL DataGlove

The DataGlove is a device which allows the flexion and extension of the user's fingers to be tracked, as well as the position and orientation (x, y, and z., roll, pitch and yaw) of the user's hand in three-dimensional space. Optical flex sensors mounted on the back of a black lycra glove measure flexure in the two inner most joints on each finger and the thumb. The device translates these flexing motions into electrical signals which allows each movement of the fingers to be recorded. In turn, this also allows various gestures to be recognized by the host computer as command input.

VPL DataSuit

The DataSuit is a whole-body input device which utilizes the same optical flex sensor technology that the DataGlove uses, but measures a much larger number of joints on the human body moving within a 10 by 14 foot area around the control system.

Applications for the DataGlove and DataSuit are being developed in such area's as the assessment of various human performance tests, range of motion measurements for motion measurements for medical diagnostics, simulations for training of personnel in hazardous or remote environments as well as being one of the emerging standard I/O devices for computer animators.   The DataSuit and DataGlove are produced and distributed by VPL Research of Redwood City, California.

Dexterous Hand Master

The Dexterous Hand Master is a high precision exoskeleton which measures the joint motions of the human hand. It is being successfully used to control dexterous robot hands such as the UTAH/MIT hand. Other uses include dynamic measurement and rehabilitation of the human hand, virtual environment control, and human factors research.

The DMH system collects joint angle data on 16 finger joints, as well as radial and ulnar deviation of each finger in real time, through the use of Hall Effect sensors and an AT compatible computer. A Hall Effect sensor is a small semiconductor which changes its signal output voltage in proportion to the magnetic field it is experiencing.

"Talking Glove"

Developed by a graduate student in the Department of Electrical Engineering at Stanford University, this device is similar to the **DataGlove** and **Dexterous Hand Master** and **PowerGlove** in that it enables hand movements to be monitored by a computer. Specifications of the glove itself have not been released as patent applications are still pending approval.

Mattel Power Glove

As opposed to using an optical flex sensing technology to track the flexion and extension of the user's fingers, this device introduces a clever and inexpensive approach to achieving the same effect. The system utilizes a 3.5 inch strip of polyester, covered with a thin coating and a conductive ink. When the user bends his/her fingers, the resistance of this flex sensor changes. This change in throughput can be quantified and used to determine the amount of bend in the user's fingers.

Position and orientation tracking is accomplished through the use of sonic positioning system.

Dimension 6

Produced by **CIS Graphics**, the Dimension 6 is a force/torque graphics control ball. This device utilizes infrared optical sensor technology which allows manipulation of graphically simulated three dimensional objects with one hand. By twisting the ball and applying soft directional movements, the object is manipulated with ease over six degrees of freedom.

2-1.9   Image Generation

The generation of virtual environments at interactive frame rates is a computationally intensive task. This requires the use of specialized 3-D graphics hardware capable of rapidly generating a large number of polygons along with minimal transport delay. Although there are a wide range of 3-D graphics workstations available on the market (as well as in use with custom designed virtual reality system) the only commercial virtual reality system (RB-2 by VPL Research, Inc.) requires the use of highend workstations as the image generators. In order to provide high frame rates, one workstation is required for each eye.

Using the **Silicon Graphics IRIS 4D-80GT** workstations, virtual environments comprised of approximately 1400 simultaneously visible polygons can be rendered at a rate of approximately 10 Hz. This produces images which are "cartoonish" in appearance as well as an acceptable frame rate. Nonetheless, on the RB2 system, the visual sensation of presence within these virtual environments is stunning.

The **Autodesk Cyberspace** system is a monumental achievement in that the graphics engine is not a pair of highend graphics workstations, but a single **80386** computer outfitted with a pair of highspeed graphics processor boards from **Matrox Electronic Systems** of Toronto, Ontario. This system is capable of generating images at a rate of 10 frames per second. There is a noticeable flicker, but it is still quite acceptable.

It is obvious that more sophisticated hardware is required to allow more complex environments to be displayed at realtime frame rates. Such power is not far away. For example, **Intel** recently introduced their new 80860 (**i860**) microprocessor. This one million transistor, 64 bit microprocessor is the fastest floating point engine available. The 40 MHz version is capable of peak execution rates at 80 MFLOPS, 33 VAX MIPS and 85,000 Dhrystones, A dedicated 3-D graphics unit, also on the chip, can generate approximately 21 million Gouraud-shaded pixels per second (50,000 100 pixel triangles, Gouraud-shaded, transformed and Z-buffered).

Division, LTD. of the United Kingdom has integrated the i860 microprocessor into a promising new virtual reality system which is based on massively parallel transputer hardware. The system is modular, allowing the addition of more processors as task complexity or the number of I/O devices increases.

It has been indicated by numerous research sites in that the **DECstation 5000** graphics workstation is becoming the standard image generation platform for the next generation of virtual reality systems.

2-1.10   Optics

Most head-mounted virtual environment display systems utilize small LCD arrays (approximately 3 inches) as image sources, one in front of each eye. In order for the user to receive a wide angle, stereoscopic view of the images displayed, a special optical system must be integrated into the headmount.

Almost all of the head-mounted displays currently used in virtual reality research utilize the **LEEP** (Large Expanse, Extra Perspective) optical system from **POP Optix Labs**

of Waltham, Massachusetts. This optical system provides a 120° horizontal field of view, binocular overlap of 90° and impressive depth cues. This firm is the sole producer and supplier of this optical system.

The use of miniature cathode ray tubes for displaying images requires an optical system considerable different than those described above. The small size of these CRT's, as well as their high luminosity, enables placement on the mounting system such that it does not occlude the visual path of the user. In order to display these images, the visual information must be passed through a set of relay optics to be directed into the user's eyes. This is usually accomplished through the use of fiber optics, prisms of varying geometries, combiners or beamsplitters.

There are benefits to "projecting" the images to the user. By reflecting the visual information off of a partially silvered mirror (beamsplitter) or tilted mirror combiner and into the user's eyes, images can be "overlayed" on the real environment. By focusing these images at optical infinity, minimal or no refocusing is required when attention is switched between these synthetic overlays and the true surrounding environment.

A head-mounted display system developed at the **University of North Carolina** actually combines various aspects of both of the methods described above. Two LCD arrays are mounted horizontally approximately the level of the user's eyebrows. Half silvered mirrors at a 45° angle are mounted below the LCD, with basic magnifying optics in between. This enables the user to view the images as overlays on the existing environment.

2-1.11 Speech Recognition

Aside from facial expressions and gestures, speech is the most basic form of human communication. Studies have also shown that it is our most rapid form of communication. A considerable amount of research has conducted in the area of electronic voice recognition in an attempt to enable more intuitive methods of interacting with a computer.

Nearly all of the voice recognition systems currently available operate through pattern recognition. Initially, a user trains a system by providing voice samples of words and phrases which are domain specific. These voice samples are sent through a digital signal processor and stored for later use by pattern recognition algorithms.

A tremendous amount of research work in this area has been conducted by, or for, two major Wall Street financial firms: **Shearson Lehmann Hutton** and **Rueters**. This was in response to the need to find solutions to data entry problems, particularly in trading.

considerable interest has been shown by the medical community as well as to facilitate the timely and accurate transcription of patient data and medical imaging evaluations.

Kurzweil Applied Intelligence Systems of Waltham, Massachusetts has developed an impressive set of such systems for use in the medical community. One system, known as **VoiceRAD**, allows radiologists to dictate into a phone-like handset while evaluating-rays. This dictation is translated from voice to text automatically.

Articulate Systems of Cambridge, Massachusetts have developed a voice recognition system, known as the **Voice Navigator**, for use with the **MacIntosh** computer. with this device, spoken commands can execute any function normally executed through the use of they keyboard or mouse. Although this device cannot be used to dictate text directly into the computer, it is capable of executing any command or menu selection. There is also a utility which allows custom applications to be developed. It has been indicated that this device has been used in conjunction with the Autodesk virtual reality system.

Researchers in the Human-Machine Interface group at MIT's **Media Lab** are conducting research into the development of computer lip reading. Such a system would augment speech recognition devices in distinguishing between similar acoustic signals by analyzing lip positions.

There are numerous additional companies involved in the development of voice recognition systems. These involved in the development of voice recognition systems. These include:

* AT&T
* IBM
* Dragon Systems Inc., Newton, Massachusetts
* Digital Equipment Corporation
* National Institute of Standards and Technology
* Dragon Systems, Newton, Massachusetts
* The Voice Connection, Irving, California
* Voice Control Systems, Dallas, Texas
* Verbex Voice Systems, Cedar Knolls, New Jersey
* Voice Recognition Technologies, Inc., Rockville, Maryland
* Covox, Inc., Eugene, Oregon
* Votan, Inc., Fremont, California

2-1.12   Technical Specifications

The technical specifications for a number of virtual reality devices are provided below, which are currently used in this industry.

2-1.12.1   Space Isotrak

–   Produced by **Polhemus Inc.**, Colchester Vermont, the 3Space Isotrak is a magnetic positioning system which determines the sensors position (x, y and z) as well as orientation (roll, pitch and yaw) in three dimensional space.

Position Coverage:

The system will provide the specified accuracy when the sensor is located with 30 inches of the source. Operations with source to sensor separations up to 60 inches is possible with reduced accuracy.

Angular Coverage:

The sensor is all altitude.

Static Accuracy:

Position: .13 inch RMS from 4 to 15 inches source to sensor separation. From 15 to 30 inches positional resolution degrades linearly to 0.25 inch RMS at 30 inches.

Angular: 0.85o RMS to 30 inches.

Position: 0.09 inch RMS to 15 inches. From 15 to 30 inches, positional resolution degrades linearly to 0.18 inch RMS at 30 inches.

Angular: 0.35o RMS to 30 inches. Output is at 60 updates/second at 19.2K Baud, Binary format.

Quite Mode:

Angular and positional resolution is improved by a factor of 3 over Normal Mode resolution. Output is at 28 updates/second at 19.2K Baud, Binary format.

Interfaces:

Serial RS-232C, Selectable Baud rates, ASCII or Binary format.

Operating Environment:

Large metallic objects, such as desks of cabinets, located near the source or sensor may adversely affect the performance of the system.

Power Requirements:

90-130 VAC, or 180-250 VAC; 47-63 Hz., 30 W.

2-1.12.2   DIMENSION6

DIMENSION6 is a patented force/torque graphics control ball produced by **CiS Graphics** of Westford, Massachusetts that provides users with natural, intuitive manipulation of screen objects on three dimensional graphics systems. DIMENSION6 utilizes infrared optical sensor technology that allows complete control in real-time of graphically simulated three-dimensional objects with one hand, while affording the user total concentration on the screen object. Real-time object manipulation is achieved by means of DIMENSION 6's internal architecture which provides six degrees of object manipulation access. In all, DIMENSION6 contends with six total "degrees of freedom" which include the X, Y, and Z coordinates, roll azimuth and elevation.

Operating internally via light emitting diodes, the optical sensors measure forces and torques applied to the sphere by the user's hand. No strenuous movement of the user's hand is required; the user simply applies appropriate, soft, directional movements to the control ball for object manipulation. The associated analog values are then converted to digital format and sent via RS-232 interface to the host workstation or computer.

Weight:	1.5 kg
Sensor:	Optically working force/torque sensor with 6 degrees of freedom; 8-bit resolution per axis; analog circuits in SMD technology, integrated in a sphere.
Functions:	Keypad with 8 buttons for user defined functions; 3 buttons for local operating mode selection; 1 reset button.

Data Transfer Interface: RS-232 with variable baud rates, daisy chain function.

Diagnostics: Built in diagnostics firmware.

Power: 220V/50 Hz. and 110V/60 Hz.

2-1.12.3   Dexterous Hand Master

As described earlier, the Dexterous Hand Master by **Exos, Inc.** is a high precision exoskeleton which accurately measures three bending motions of each finger in addition to radiallunar deviation (side to side motion).

Hand Size: 5th percentile Female to 95th percentile Male.

Joints: Up to 20 joint angular positions

Angular Resolution: ~0.5 degrees

Exoskeleton Weight: ~11.0 ounces

Power: 110 volts AC, less than 2 Amps

Temperature: 25o C, ± 10o C.

Data Acquisition Rate: Each channel sampled up to 100 times per second.

Processor Type: AT compatible microcomputer

Output: RS-422, RS-232

Max. Output Rate: 100 Hz (RS-422)

Recording Rate: 1-100 Hz, or by event.

2-1.12.4   VPL DataGlove

As described earlier, this is a patented computer input device which converts hand gestures into computer readable form. The DataGlove consists of a thin lycra glove with fiber-optic joint angle sensors on the thumb and fingers and a **Polhemus** magnetic

positioning device on the back of the hand. The microprocessor-based control unit acquires data from the DataGlove and transmits it to the host.

Joints measured: Metacarpophalangeal joints of five fingers, the interphalangeal joint of the thumb, and the proximal interphalangeal joints of the other four fingers.

Angular resolution: 1 degree

Hand Orientation: 3 degrees of freedom, orientation and attitude.

Position and attitude measurement accuracy (at 15 inches) 0.13 inches RMS
0.85 degrees RMS

Data acquisition rate: 60 times per second (Note: A high accuracy mode yields 0.03 inches RMS accuracy at 28 samples per second)

Data communications interface: RS-232 and RS-422 (user selectable from 300 to 19,200 baud).

Data transfer rate: 30/second, 60/second or on demand.

Operating temperature: 10° C to 50°C at relative humidity of 10% to 80%, noncondensing.

Operating Environment: The six degree of freedom tracking system should not be used in the presence of ferrous objects or strong electromagnetic fields.

Power requirements: 100 VAC, 50-60 Hz, less than 100 watts.

2-1.12.5 EyePhone

Produced by **VPL Research** of Redwood City, California, the EyePhone is a head-mounted display device that utilizes miniature LCD arrays and a unique set of stereoscopic wide angle optics. The Eyephone system totally occludes the view to users surrounding environment. Only the images on the liquid crystal displays are visible.

Weight:	EyePhone Goggles:	2.25 lbs.
	w/counterweights straps:	4.25 lbs.

Field of View:

Horizontal - per eye:	80 degrees.
Horizontal - both eyes:	100 degrees.
Vertical:	60 degrees.

Resolution per Eye:

(primary colored pixels) 360 x 240

Tracking Accuracy:

Position:	0.13 inches RMS (to 15 inches). 0.25 inches RMS (at 30 inches).
Orientation:	0.85 degrees RMS (to 30 inches).

Video Inputs: Left and right inputs, NTSC video.

Data Communications Interface: RS-232 and RS-422 (user selectable from 300 to 19,200 baud).

Data Transfer Rate: 60 times per second.

Operating Temp: 10o C to 50oC at relative humidity of 10% to 80% noncondensing.

Operating Environment: Not to be used in the presence of metallic objects of magnetic fields.

Line Fuse: 2 Amp, 250 Volt Fast blow.

2-1.12.6   US Air Force Super Cockpit

The display system developed by researchers at **Wright Patterson Air Force Base** in Ohio utilizes an entirely different approach.

Developed for a system known as the SuperCockpit, researchers have designed a miniature cathode ray tube which measures .25 inches in diameter on which an extremely bright 750 x 750 pixel image can be projected. Optical components magnify, collimate and projects the images directly into the user's eyes.

Such a system is unique in that user can see through the images into his normal visual field. These enhancements are commonly referred to as virtual overlays.

2-1.12.7   Broderbund U-Force

This is a new input/control device developed for the **Nintendo Entertainment System**. This system is rather unique in that, through the use of infrared tracking technology, there is no physical contact between the user and the device while in use. The system is capable of tracking motion, velocity and relative position of the user's hands and using this data to control characters on the screen.

The device is approximately the size of a collage text book and the top folds open away from the user. When activated, the device generates a 3-D infrared light grid which monitors all movement within its area of coverage. The use then moves his/her hands in within the grid (punching motions, thumb movements for trigger control, etc.). For those users who would prefer the use of standard interfaces, the system also includes firing handles and a T-bar. It has been indicated that the exact method of operation is proprietary.

2-2.0   Review of Baseline Cupola Workstation (CWS) Functions

This section reviews crew functions and tasks to be accommodated in the cupola workstation.

2-2.1   Cupola Control Functions

2-2.1.1   Control Capabilities

The following control capabilities will be available to a crew member while performing and viewing associated operations from a cupola:

-    Space Station manipulators (including the servicing facility manipulator or robotic devices except the JEM manipulator);

- Station manipulator transporter (with or without manipulator attached);
- Telerobotic servicer (whether attached to a manipulator, transporter, or OMV);
- OMV piloting (including piloting of any unmanned commandable vehicle within the Command and Control Zone (CCZ);
- Servicing facility enclosure and all operations within the servicing facility;
- External video cameras and lights (including servicing facility cameras and lights) and internal (cupola) video monitoring;
- Any visual alignment, range, or angle sighting devices;
- Internal and external voice communications (including C&WS);
- Systems control functions available through DMS access;

2-2.1.2 Exterior Cover(s)

Exterior cover(s) will be automatically or manually controlled from inside the cupola.

2-2.1.3 Thermal Conditions

Thermal conditions within the cupola will be controlled to pressurized volume specifications.

2-2.1.4 Safety Features

Remote safety verification of controllable unmanned vehicles operating in the Station.

2-2.1.5 Dynamic Control

Exercise dynamic control of all unmanned vehicles within Station CCZ. Provide backup navigational and maneuver targeting for all manned vehicles operating within CCZ.

2-2.1.6 CERS Control

Capability will be provided for a crew member in the cupola to control activation, deployment, and transfer of the CERS (Crew/Equipment Retrieval System).

2-2.1.7 External Element Control

Each cupola will provide the capability for two crew members, side-by-side to view and control external elements while facing in the +/-X or the +/-Y directions.

2-2.1.8 Additional Derived Control Functions:

- Window defogging control (31000 par 12.2.8.3.3.2 says "Defog and contamination protection will be provided.");
- Cupola interior lights control (On/Off/Brightness);
- Cupola interior window shades control (Open/Close totally or partially);
- Reposition displays and controls to different locations within cupola;
- Reposition crew restraints to different locations within cupola;
- Control cupola ventilation control;
- Stow, destow, mount, activate, and operate optical sighting devices;
- Stow, destow, use binoculars;
- Control of space to space, space to ground, and intra station voice communications (select channels, adjust volume, etc.);
- Stow, destow, mount, activate, and operate cameras.

2-2.2 Direct Viewing Functions

2-2.2.1 Direct Viewing Requirements (via windows):

1) General viewing will be provided to monitor docking and berthing;
2) General viewing will be provided for proximity operations;
3) General viewing will be provided of the Orbiter cargo bay for cargo handling;
4) General viewing of MSC movement, payload and SSPE attach points, the alpha joints, and servicing areas will be required;
5) General viewing of EVA/EEU and CERS operations will be provided;
6) General viewing will be provided for IVA-controlled robotic maintenance operations;
7) A window will be provided for use of coarse optical alignment sight sextant/attitude information tasks;
8) Windows and/or cupolas used in conjunction with proximity operations and/or MSC operations will be sized to permit simultaneous use by two crew members.

2-2.2.2 Direct Crew Visibility

Direct crew visibility is required for all proximity operations within 60 meters (200 feet) of the Station. For separation (translation) maneuvers resulting in a trajectory with closest point of approach less than 300 meters (1000 feet), the crew must have either direct visual contact or electronic acquisition to ensure safe clearance. There is no requirement for visual contact with the vehicle for a gradually increasing relative motion separation trajectory at ranges outside of 60 meters (200 feet). For nighttime visibility these

requirements may be achieved through the use of Station floodlights if vehicle reflectivity is sufficient; otherwise, through appropriate use of vehicle running lights.

2-2.2.3 Approaches

The Station configuration and design should support nominal final approaches to the Station from the plus or minus V-bar (along the velocity vector) direction.

2-2.3 Indirect Viewing Functions

2-2.3.1 CCTV System

The CCTV system (with zoom capabilities) will be provided to augment direct window viewing of specific areas and when required to accomplish specific tasks. The CCTV system is a Space Station capability that will be available to crew with minimal delay. Indirect viewing requirements follow:

1) Specific viewing of the docking interface will be provided;
2) Specific viewing of MSC operations will be provided (berthing/deploying interfaces, grappling interfaces);
3) Specific viewing of EVA/EEU and CERS operations will be provided (detail of EVA operation and constant sighting of EVA crew member);
4) Specific viewing of all work areas will be provided;
5) Specific viewing of robotic maintenance operations will be provided;
6) Specific viewing will be provided for remote inspection of any point on the external Space Station structure. This capability may be achieved through the use of fixed cameras, EVA hand-held cameras, and/or cameras mounted on the MSC;
7) Specific viewing will be provided of all payloads, platforms, OMVs, OTVs, and free flyers and their attach points during all operations;
8) Indirect viewing will include viewing of future additions to the Space Station, or the capability to expand to include direct viewing of the expansion elements for inspection purposes.

2-2.3.2 Attachment Identification

Positive identification of successful attachment of any vehicle to the Station will be provided to the SSF crew.

2-2.3.3 Monitoring Devices

The MT will consist of monitoring devices for determining the position of the transporter on the truss.

2-2.3.4 Warnings

– The SSF IVA and EVA crew will be automatically notified of the onset of an EMU warning.

2-3.0   Overview of the Proposed Baseline BP/VCWS

2-3.1   Definition of Body-Ported Virtual Cupola Workstation (BP/VCWS)

The Body-Ported Virtual Cupola Workstation BP/VCWS is an advanced component of the space station integrated workstation system. Its primary purpose is to provide the cupola crew with flexible virtual display workstations that possess intelligent control capability, and therefore, the ability to perform all cupola control and interfacing functions. Though this device is designed for the dedicated use of cupola crew members, it has the function and flexible interfacing to allow crew members to use at several locations in the space station environment.

The BP/VCWS system (Figure 2-1) consists of three main assembly modules:

1) a head-mounted assembly which provides the virtual display and communication interfacing;
2) a Body-Ported High Performance Controller (BP/HPC); and,
3) a special-purpose command/control input module with several input devices, such as multi-degree-of-freedom master arm/hand controller and special keyboard.

The BP/VCWS is an advanced component of the SSF Multi-Purpose Application Console (MPAC) system, which is the electric core of the hardware components of the crew workstation. The MPAC provides a multi-functional interface into all operation monitoring, controlling, training, testing, caution and warning functions, and other crew services provided through the DMS.

The BP/VCWS also is a part of the space station integrated workstation network system, and will provide the crew services through the DMS system. The BP/VCWS will employ, to the maximum degree, all common MPAC Orbit Replaceable Unit (ORU)

components for easy maintainability.

2-3.2　Basic Functions of BP/VCWS

The BP/VCWS is an advanced human interface device that will provide the following functions:

1) Cooperative interaction with other workstations contained within the SSF;
2) Cooperative interaction and control with the Space Station MSC and the NSTS RMS;
3) Cooperative interaction with ground stations;
4) Interface with all Space Station DMS and the NSTS data system services and control functions;
5) The man/machine interface and decision support for control of both teleoperated and telerobotic operated systems operations.

2-3.2.1　Single Person Operation

Single Crew Member Operation - Where practical and cost effective, BP/VCWS operation and scheduled maintenance, station-provided user servicing operations, and contingency response operations will be designed for performance by a single crew member.

2-3.2.2　Override

The BP/VCWS operator must be able to override or reverse any automatic safing or switchover capability of functional paths. Overrides of automated functions affecting system safety will be two-step operations with positive acknowledgement required of the initiator.

2-3.2.3　General Function Requirements

The proposed BP/VCWS will be configured to meet the control and interfacing functions and performance requirements of the workstation cupola. These functions can be classified into six categories:

1) Operation Monitoring in the Space Environment;

2) Teleoperated and telerobotic systems control;
3) Caution and Warning Systems (C&WS);
4) Training and simulations;
5) DMS System Testing and Diagnosis;
6) Crew Services.

2-3.3 Significance of Using BP/VCWS

The proposed BP/VCWS belongs to the Space Station Advanced Automation Program, and will meet the evolutionary requirements of the future space station cupola workstation operations.

This multi-disciplinary high-tech device represents the state-of-the-art computing science. It employs the advanced "virtual world" display and intelligent control technology, which enables the cupola crew operators to perform sophisticated space operations with higher efficiency and lower risks. It will eventually replace the "hard console display" workstation which is being designed under the present space station development program.

Specifically, the proposed BP/VCWS will have the following tools directly accessible to the crew users:

1) 3-dimensional, stereoscopic, wide-field-of-view (WFOV), and high resolution color display capabilities;
2) head-tracking mechanism equivalent of multiple displays;
3) Knowledge-based System (KBS) controller that features intelligent control functions;
4) optional portable special keyboard and arm/hand controllers.

The major advantages of using BP/VCWS are to:

1) improve human-system interface for cupola workstations;
2) replace bulky and redundant permanently installed hard console workstations;
3) increase flexibility and portability over the "slidable" console concept;
4) simplify evolution of crew workstations;
5) potentially reduce costs of system evolution.

2-3.4   Feasibility of BP/VCWS Development

The development of the proposed BP/VCWS is directly related to the rapidly growing and maturing technologies in the fields of high performance microcomputers, graphics display, artificial intelligence, and control systems. The preliminary ADCACS Phase One study recommends that the development and application of BP/VCWS is feasible and practical, with acceptable technical, schedule, and cost risks.

The development of BP/VCWS is also directly related to the following very recent on-going projects which provide solid backgrounds for establishing the BP/VCWS plan.

1) VIVED workstation system and EVA Helmets (NASA/Ames/JSC)
2) Space Station Advanced Automation Program (NASA)
3) Integrated Helmet and Display Sight System (IHADSS, Honeywell)
4) Visually Coupled Airborne Systems Simulator (VCASS, Wright-Patterson Air Force Base)
5) Fiber-Optic Helmet Mounted Display (FOHMD, Williams Air Force Base)
6) "Super Cockpit" System (Air Force)
7) "Pilots Associate" System (McDonnell Douglas)
8) C Language Integrated Production System (CLIPS, NASA/JSC)

2-4.0   Baseline BP/VCWS Design Requirements

The purpose of defining the design requirements and configuration items for the BP/VCWS is to provide a practical approach for the construction and delivery of a flight-qualifiable device that satisfies the cupola performance requirements described in Section 2-3.0. Several issues regarding this purpose will be addressed in this section.

2-4.1   BP/VCWS Architecture

2-4.1.1   Modular in Design

The proposed BP/VCWS is modular in design and will be capable of being evaluated and reconfigured to meet future operation control requirements of teleoperated and telerobotic systems through modular attachment of unique controls, displays, electronics,

and high level processors.

The proposed BP/VCWS will consist of:

1) a head-mounted multi-media assembly module;
2) a body-ported high performance controller (BP/HPC) system assembly;
3) a special purpose keyboard and master arm/hand command/control input device module.

The head-mounted module has a 3-dimensional, stereoscopic, high resolution, multi-color display with head tracking device, voice recognition/communication equipment. The BP/HPC system assembly is a microprocessor-based, high speed computing device that is comprised of a high-speed display/image/graphics rendering processor and a knowledge-based intelligent controller which provides control functions and decision support for the cupola operations. Also, the BP/HPC includes a network controller that provides a gateway to incorporate the main SSF DMS system for various crew services. All components in the BP/HPC module are assembled of Orbital Replaceable Units attached to the SSF structure in the cupola. The interfaces to other space station workstation systems are provided by a connecting cable to the SSF DMS.

2-4.2 BP/VCWS Configurations

This section provides the qualitative and quantitative requirements for the individual configuration items of the BP/VCWS system and characteristics of these items.

2-4.2.1 Head-mounted Assembly Module

The head-mounted assembly module provides the direct human/system, multi-media interfacing to the cupola crew operator with "virtual reality" interfacing capability.

2-4.2.1.1   Head-Mounted Assembly Functional and Performance Requirements

The BP/VCWS head-mounted assembly, in conjunction with the BP/HPC and with the DMS USE control software, will provide the following functional capabilities:

1) Display of the standard ASCII alphanumeric character set with fonts and extensions as defined in the User Interface Requirements Document (MDC

H4261);

2) Graphical display of 2D (two dimensional) and 2D representations of 3D (three dimensional) objects, with capability of upgrading to true 3D, stereoscopic, wide-field-of-view, high-resolution, and multi-color virtual displays;

3) Interactive animation of 2D objects with capability of upgrading to real-time, interactive, true 3D, solid simulation and imagery display;

4) Display of National Television Standards Committee (NTSC) full motion, multi-color video views;

5) Multi-scale black-and-white images display of up to 16 gray-scales, and multi-color display capability with various combinations of resolutions, and up to 256 colors;

6) Multiple windows, with varying size display capability, in accordance with the User Interface Requirements Document (MDC H261);

7) Hypertext display capability within any one window. (mixing of video, alphanumerics, and graphics);

8) Display freeze control/release capability;

9) Audio annunciation with simultaneous visual information display for reporting information such as emergency conditions and user prompts for workstation operation;

10) Voice recognition capability for control and command input and voice communications within the SSF and with ground station;

11) Capability of 2D representation of 3D graphics with zoom and rotation;

12) A head-tracking mechanism equivalent of multiple displays.

2-4.2.1.2 Head-Mounted Assembly Configurations

The head-mounted module will be composed of the following items:

1) The helmet frame;
2) Display screen/monitors;
3) Wide-field-of-view (WFOV) optical lenses;
4) A head tracking device;
5) Communication and voice recognition device, including microphone(s) and speaker(s);
6) Other accessories and utility support for illumination, ventilation, insulation, and mounting;
7) An Interface cable to the BP/HPC;

2-4.2.1.2.1 The Helmet Frame

A helmet frame is the housing for all head-mounting display, headtracking, and voice/ communication devices. It will be made of high strength, light-weight alloy. The detailed size and mechanical design data are to be provided.

2-4.2.1.2.2 Display Screen/monitors

A dual-screen (one for each eye), color monitor will be installed as a major display device that supports the 3D stereoscopic display capability. It will support all display functions specified above. Other performances of the monitor include the support of WFOV display and high resolutions. The resolution will meet the minimum human factor requirements for comfortable settings and long lasting work period.

The monitor will be safe and light weight, will have reliability and will be designed for easy maintenance and orbital replacement.

2-4.2.1.2.3 Wide-Field-of-View (WFOV) Option

The head-mounted assembly will have an option of wider field visibility than the normal display. This option is very useful to the telerobotic operation, monitoring, and other EVA operations. This function will be implemented by installing a Wide-Field-Of-View lens, together with the nonlinear compressed image rendered to the display monitor.

2-4.2.1.2.4 Head Tracking Device

A head-tracking mechanism will be installed in the head-mounted assembly, which provides multiple virtual display directions according to the head position. The model of the head tracking device is based on the NASA/Ames VIVED workstation.

2-4.2.1.2.5 Communication and Voice Recognition Devices

This configuration includes microphone(s) and speaker(s) installed in the head-mounted assembly. It provides the audio communication and voice input capabilities to the crew operator. Also, this built-in audio system will provide annunciation of all audible caution and warning messages and tones.

2-4.2.1.2.6   Interface Cable to the High Performance Controller (BP/HPC)

The interfacing between the head-mounted assembly and the BP/HPC is provided by a combined fiber-optic/electronic cable. It will provide display image transmission and voice recognition/audio communication channels. The required bandwidth for the video image channel is 100 MB/second.

2-4.2.1.2.7   Flexibility

Due to operational complexity and varying schedules of the cupola crew operator, the head-mounted assembly will be designed to be easily put on and off to maintain flexibility.

2-4.2.1.2.8   Bulk Weight

It is initially expected that the total weight of the head-mounted assembly will not exceed 20 pounds. Final weight will be determined later.

2-4.2.2   Body-Ported High Performance Controller (BP/HPC) Assembly

The Body-Ported High Performance Controller is the central computing device that controls and manipulates all functions that the BP/VCWS performs. It also provides a network gateway to the main SSF DMS. It is the electric core of the BP/VCWS. Though it is designed as a high performance MPAC, BP/HPC is considered as a component of the MPAC device, and is consistent with hardware and software design procedures, and standards. The BP/VCWS will be made compatible to the hardware module and will be able to replace the general purpose MPAC ORUs used by other SSF workstation systems.

2-4.2.2.1   BP/HPC Characteristics - Intelligent Control and Decision-Making port

The major significance of the BP/HPC system is the capability of artificial intelligence. It employs a Knowledge-Based System (KBS), which is a subset of the SSF Advanced Automation. It provides intelligent control and decision support to the cupola crew operator for normal and emergent cupola operational activities.

2-4.2.2.2   Modular in Design

The BP/HPC is an independent firmware module. Its hardware and software will be capable of being reconfigured and upgraded to meet the evolutionary control and interfacing operation requirements of the BP/VCWS, particularly the teleoperated and telerobotic control system operations.   This will be accomplished through modular attachment of unique controls, displays, electronics, and high power processors.

2-4.2.2.3   BP/HPC Assembly - Functional, Interfacing and Performance Requirements

The following paragraphs define the functional, performance, and interfacing requirements of the BP/HPC .

2-4.2.2.3.1   Teleoperated and Telerobotic Control

Display output capabilities and operator input functions will be provided to control and monitor all distributed systems, and teleoperated systems including the Mobile Servicing System, Free-Flyers, and the Flight Telerobotic Servicer (FTC).   These capabilities will include:

1)   Multiple Workstation Monitoring

The BP/HPC will provide control to monitor the telerobotic activities by BP/VCWS. This monitoring capability is also provided by other SSF workstation systems.

2)   Transfer of Control

The BP/VCWS operator will be able to transfer control of the teleoperated or telerobotic system to a cooperating workstation in a safe and controlled approach.

3)   Master Arm/Hand Controller

Control functions for at least one, 6 degrees-of-freedom master arm/hand controller will be provided by the BP/HPC.   Control capacity for evolutionary installations of additional masters/manipulators will be provided.

4)   Master EOAT Controller

The control of at least one multiple degrees-of-freedom End-of-Arm-Tooling (EOAT) controller will be provided by the BP/HPC. The EOAT Master controller will have the capability of force-reflective one degree-of-freedom control (parallel jaw gripper), position control, hybrid force/position control, and resolved rate control.

2-4.2.2.3.2   Control of Indirect Video Viewing System

An on-board video viewing capability will be provided to the BP/VCWS by the C&WS and DMS to enhance direct window viewing of specific areas, and when required to accomplish specific tasks. The real-time on-board video capability will be available to the BP/VCWS. The BP/HPC will be capable of providing this control function. Indirect viewing services provided are:

1)   Specific viewing of the docking interfaces;
2)   Specific viewing of the MSC operations (berthing/deploying interfaces, grappling interfaces);
3)   Specific viewing of EVA operation details and constant sighting of EVA crew members);
4)   Specific viewing of internal quarters and work areas;
5)   Specific viewing of robotic maintenance operations;
6)   Specific viewing for remote inspection of any point on the external station structure. This capability may be achieved through the use of fixed cameras, EVA hand-held cameras, and/or cameras mounted on the MSC;
7)   Specific viewing of all OMV's, payloads and their attachment points as well as free flyers during all operations;
8)   On board video viewing requirements will be met by adequate cameras with pan, tilt, and zoom capabilities appropriate to fulfill the requirements of 3.2.1.1 of JSC 31020;
9)   Indirect viewing will allow for future additions to the Space Station.

2-4.2.2.3.3   Control of Caution and Warning Service (C&WS)

This control function will be provided and implemented by BP/HPC with interfacing to the DMS. It will support control of the dedicated video/audio annunciator for reporting emergency, caution and warning alerts. This service will include:

1) Procedures to be invoked by application programs for purposes of monitoring system data for abnormal conditions that may require crew notification;

2) Applications for providing the control and monitoring function between the crew member and Space Station payload systems. Where control and monitoring are done automatically, this service provides the appropriate feedback and/or C&WS.

2-4.2.2.3.4   Control of Voice Recognition and Communications

Voice activation of control functions will be utilized for more reliable and more productive operations. When voice activation is used, the following functions will be provided:

1) Discrete utterance input;

2) Continuous speech recognition input: Continuous speech recognition is not necessarily required for baseline function requirements. However, the BP/HPC system will have the capability of providing evolutionary installation and utilization of discrete and continuous voice recognition, which can be more naturally, successfully, and eventually developed;

3) The intelligent control of the BP/HPC will provide system protection against voice activation caused by casual conversation, noise or system errors in recognition;

4) Internal intercom transfer and pickup service: Internal intercom transfer and pickup service also will be controlled by the BP/HPC. The audio/video system will be synchronized. The audio system must be able to carry all speaking voice frequencies required for satisfactory intelligibility. The audio system will not preclude the use of speech recognition and encrypted channels.

2-4.2.2.3.5   Real Time Monitoring, Command, and Control

The capability of real time command and control, calling up status displays and conducting command and control operations for Space Station flight safety critical systems and designated payloads will be provided by the BP/HPC intelligent control software.

2-4.2.2.3.6   Control of BP/VCWS Crew Service Functions

The BP/HPC, in interaction with the SSF DMS through interfacing, will support the following crew service functions:

1) Training;
2) On-line Help;
3) Word Processing;
4) Electronic Mail;
5) Document Retrieval;
6) Text Processing;
7) Calculator;
8) Spreadsheet.

2-4.2.2.3.7   Data Processing and Printing

The BP/HPC, also through DMS, will provide the means for data acquisition, retrieval, processing, storage, display, and printing. An interface control to a printer/plotter for the output of real-time generated data, stored information, or display screen hardcopies will be provided.

2-4.2.2.3.8   BP/VCWS System Initialization and Shutdown

The BP/HPC will have the capabilities of controlling the entire hardware and software systems, including initialization and shutdown of any components of the BP/VCWS and other space subsystems.

2-4.2.2.3.9   BIT/BITE

Built-in-Test-Equipment (BITE) will be provided for all BP/VCWS components for status reporting, fault detection, fault isolation, and maintenance and repair support of all distributed and safety critical systems.

2-4.2.2.3.10 BP/HPC Future Upgrade and Evolution

The BP/HPC will be designed for future upgrade and adaptation for function requirements, reconfiguration and modification of the future SSF system.

2-4.2.2.4   BP/HPC Hardware and Software Configurations

2-4.2.2.4.1   General Scope

The BP/HPC is the central interfacing and control unit. The BP/HPC is comprised of an advanced high-speed central processor, the knowledge-based intelligent control and conventional control software, and high performance storage units (ROM, RAM and secondary storage devices). It plays a key role in the display, control, and interfacing functions of all cupola operations. The HPC central processor is a member of the Embedded Data Processor (EDP) family and will include a central operating system that is compatible with that of DMS's.

The BP/HPC central processor will have sufficient computing and memory capability to support crew workstation services. The BP/HPC will interface via an NIU to the DMS network.

2-4.2.2.4.2 BP/HPC Hardware Assembly Configurations

The BP/HPC has the following hardware components:

1) HPC Central Processor

BP/VCWS operations require that the BP/HPC have a high speed, high performance central processor to manipulate sophisticated multi-tasks.

The proposed baseline HPC central processor will have a 50 MHz Intel i860 microprocessor with the capability of upgrading to high speed, multi-processing microprocessors.

2) High Speed Random Access Memory (RAM)

The HPC central processor will require the use of 16 MB high access speed Random Access Memory (RAM), with 64KB Cashe capability.

The designed storage size of Read-Only Memory (ROM) for the BP/HPC central processor system will be adequate to store the operating system kernel and crucial application software. The design will make upgrading the system software much easier by only changing ROM chips.

3)   High Density, High access Speed Fixed Storage Unit

The BP/HPC system requires at least one high density fixed hard disk storage unit with capability of 200 MB. The system will also be designed to have the flexibility of installing additional hard disks to meet the storage needs.

4)   Storage Loading and Backup System

The BP/HPC system will have at least one 3.5" double-sided, high density (1.44MB) floppy disk drive and a tape drive device for software loading and backup.

5)   BP/HPC System Buses

There will be two external buses interfacing the BP/HPC to the SSF DMS system and the head-mounted assembly. The external bus connected to the SSF DMS system will meet all interfacing requirements. The design will be made compatible with, and will be contingent upon the SSF DMS bus system design.

The bus interfacing to the head-mounted assembly will provide direct human interfacing functions to the cupola crew operator. It actually connects the head-mounted assembly through a special-purpose graphics rendering processor.

The internal BP/HPC system bus will be comprised of data bus, control bus, and system input/output bus. The internal bus will be used for connections and data/control flow transfer between electronic components, and will be designed in wide bandwidth.

The BP/HPC system processor will be able to use DMS mass memory (i.e., Mass Storage Unit [MSU]), through an external bus system connection to support normal core and payload system functions.

6)   I/O Control Unit and Accessories

A Network Interface Unit (NIU) for the purpose of interfacing with SSF DMS will be embedded in the BP/HPC. Other electronic components necessary for the BP/HPC include:

a)   The special purpose graphics rendering processor (Graphics Engine);

b) I/O ports (both serial and parallel) and direct memory accessing (DMA) interfacing controller;

c) System timer, counter, and interrupt controller;

d) Other auxiliary components and accessories.

7) Compatibility and Evolutionary Growth

The baseline BP/HPC design configuration will be the same as that of the Space Station embedded processor, which is virtually Intel 80386 microprocessor compatible. It does not support multi-tasking and does not have the parallel processing capability. This configuration can merely satisfy the minimal requirement for control of BP/VCWS system operations. However, the rapid development of microprocessor technology makes it possible to upgrade the baseline BP/HPC configurations on an evolutionary basis.

2-4.2.2.4.3   BP/HPC Software Configurations

The most significant feature of the software system for the BP/HPC is that it applies Space Station Advanced Automation Technology to support intelligent control and decision support functions for the BP/VCWS. The Knowledge-Based System (KBS) will play an important role and will be imbedded in the control and monitoring software system. A detailed report on the software development considerations for the proposed BP/VCWS is included in Chapter Three, Section 4.2.

2-4.2.3   BP/VCWS Interfacing Configurations

2-4.2.3.1   BP/VCWS External Interfacing with the SSF DMS

The BP/VCWS system will have external interface which connects it to the SSF DMS. These are identified in the following paragraphs.

2-4.2.3.1.1   BP/VCWS Interface with DMS Network

DMS Fiber-Optic Distributed Data Interface (FDDI) requirements are fulfilled by the Network Interface Unit (NIU) which is embedded in the BP/HPC.

The NIU is an intelligent I/O processor and physical interface unit serving one dual redundant 100 MB/second fiber optical data network. the NIU is comprised of two

common DMS pages, a Network Interface Adapter (NIA - as described in the IBM EDP Specification 152A404) and an Embedded Data Processor (EDP - as described in the IBM EDP Specification 152A 403). As a two page set, the NIU provides layers 1 and 2 (partial) of the International Standards Organization/Open Systems Interconnect (ISO/OSI) Reference Model to implement the physical interface with the DMS optical network, with the processing capability to host the Network Operating System software.

The configuration of the hosted software is dependent upon the function of the ORU which embeds the NIU. The NIU has a throughput of 10 MB/second.

2-4.2.3.1.2   BP/VCWS Interface with TGDS

Physical connection and electrical interfacing to the Time Generation and Distribution System (TGDS) is provided to the NIU by an Electronic Industries Association (EIA) RS-422 interface. Precision time and frequency updates are available to the EDP and USE (MDC-H4192) software through the Multibus II interface. The TGDS is defined in Time Generation Unit Specification IBM 153A301.

2-4.2.3.1.3   BP/VCWS Interface with the Video Distribution System

This interface is defined by the standards in RS-170A. The BP/VCWS accepts up to three baseband National Television Standards Committee (NTSC) color encoded RS-170A television signals. Operator selection of video signals will be provided through USE software control in conjunction with the Communications and Tracking control and monitor software.

Specific closed-circuit television (CCTV) viewing requirements will be met by several camera locations both internal and external. A total of 22 moveable cameras (which can be used either EVA or IVA) with tilt, pan and zoom capabilities will be provided to meet these viewing requirements. Cameras provided by international partners or other Space Station crew users will not be considered as a part of the 22 Space Station provided cameras.

2-4.2.3.1.4   BP/VCWS Interface with the SSF Communications

Multiple, nonblocking, full duplex, audio intercommunication channels will be provided for allowing communications among all crew members, both located inside the SSF quarters and at EVA. The audio system will provide:

1) Interfaces to space-to-ground and space-to-space RF link processors, docked NSTS, international elements and wireless internal audio system;
2) Synchronization with video where required;
3) Multiple simultaneous conferring;
4) Private voice channels where required;
5) On board recording and playback of voice channels.

2-4.2.3.2   Internal Interface Configurations

2-4.2.3.2.1   BP/HPC Interface with the Head-Mounted Video Display Devices

The interface of the BP/HPC system to the head-mounted video display system is provided by a high performance, highly parallel bus interfacing cable with bandwidth of at least 100MB/second for the image rendering requirements of the animated 3D, stereoscopic, high resolution color displays. The interfacing control is made compatible with the internal bus of the BP/HPC system.

2-4.2.3.2.2   BP/HPC Interface with the Head-Mounted Audio/Voice Communication Units

Due to the relative low bandwidth requirement of audio/voice signal transmission, this interface will be provided by the conventional RS-422 serial interface which will be full duplex implementing one start bit, seven data bits, and one odd parity bit. Handshaking will include the Request to Send (RTS) signal and Clear-to-Send (CTS) signal. RS-422 equivalent interfaces may be employed with prior verification.

2-4.2.3.2.3   BP/HPC Interface with Operator Command Input Devices

The cupola crew operator input devices include master arm/hand command device and special keyboard. The interfacing of BP/HPC to these devices will be established by the RS-422 serial interface.

2-4.2.3.2.4   BP/VCWS Interface with Printer/Plotter

The BP/HPC will provide an IEEE-488 parallel interface to a printer/plotter which will provide the crew with a hardcopy of information that was generated in real time, stored in memory, or transmitted from the display screen. This information may be in graphic or hypertext formats. The desired printing speed is 400 words/second, or 8 letter-size graphic pages/minute.

The printing quality will be in true letter quality or in high resolution graphics quality. Color printing is a useful option.

This printer/plotter will be specifically assigned to the BP/VCWS crew operators, and will have the capacity of allowing other crew members to use from other space workstations through DMS networking.

2-4.2.3.3   BP/VCWS Utility Interface Configurations

2-4.2.3.3.1   Electrical Power System Interfaces with BP/VCWS

Though the electrical power system is TBD, the proposed BP/VCWS will operate and meet all system performance requirements without degradation (using electrical power supplied by the SSF power system). Electrical power will be 120 VDC. The total power consumption of the BP/VCWS system will not exceed 1100 watts.

1) BP/VCWS Head-Mounted Unit - The electrical power interface consists of a 120 VDC input as defined by JSC 30482;

2) BP/VCWS/HPC Power Interfacing System (Reference MPAC Processor Specification - IBM 152A601).

External System Interfacing

SSF DMS System

Internal Interfacing I

Head-Mounted Assembly

- Helmet-Mounted, 3D, High Resolution, Stereoscopic Color Display
- Voice Command Input
- Audio Communi-cation System
- Head-Tracking Mechanism

Body-Ported High Performance Controller (BP/HPC) Assembly

- HPC Processor and Main Storage
- High-Speed Image Processor
- Secondary Storage Unit
- Firmware Unit

Internal Interfacing II

Special Command/Control I/O Devices

- Special Keyboard
- Force-based and/or Position-based Mater
- Other Special Input Devices
- Printer/Plotter

Figure 2-1   System Overview of the BP/VCWS

CHAPTER THREE

A COMPLEMENTARY REPORT ON COMPUTER ENGINEERING RESEARCH FOR THE PROPOSED BP/VCWS

3-1.0   Organization of Chapter Three

This chapter consists of the following sections:

3-2.0   A Description of the Prototype Emulator for the Proposed BP/VCWS.

3-3.0   Marquette University Advanced Control Technology Laboratory Experiments.

3-4.0   Evolutionary Study of Hardware and Software for the Proposed BP/VCWS Design.

3-2.0   A Description of the Prototype Emulator for the Proposed BP/VCWS

This section contains a brief description of the prototype emulator for the proposed BP/VCWS, accomplished during Phase One research. It was based on a computer network system developed at the Marquette University Advanced Control Technology (ACT) laboratory.

Two aspects were taken into account when the network interconnection design was considered:

1) The resulting system must be able to conduct sub-system experiments. There may be simulations of some modules such as virtual world generation and control functions of certain components of the emulator during the design and experiments.

2) It must be able to implement several important simulations of a space operation scenario.

3-2.1   Networking for the Prototype Emulator

3-2.1.1   Network Design Requirements

For the function requirements of the prototype emulator, a variety of computing resources were needed. All these resources were not available on a single, low cost workstation. Some workstations have good graphics capability, other systems may have good I/O interfacing capability. One example is that the commonly used 386 PCs, with MS-DOS environment, have limited computing power for virtual display and graphics generation, but have good I/O interfacing capabilities. However, there are a number of commercially available I/O interfacing devices which support the MS-DOS environment and can either reside in the expansion slot or utilize the interface card in the PC's expansion bus. The Silicon Graphics IRIS Graphics Workstation, on the other hand, works under the UNIX operating system, and possesses superior graphics capability for virtual display simulation, but has less popularity in public support of I/O interfacing capability.

The appropriate methodology for simulation was to take advantage of a PC's interfacing and I/O capability and the IRIS' graphics capability, and network them together. For the proposed experiment and simulation function requirements of the prototype emulator, the network system must possess the following capabilities:

1) transfer files between systems;
2) perform remote login to the IRIS from PC's;
3) share files at different systems;
4) execute programs in real time over the network.
5) support interprocess communication over the network.

In addition, the network should allow users to utilize the resources of different systems over the network. A shared resource, like a graphics interface, would be associated with a subroutine on the UNIX environment. This must be accessible over the network so that a PC can access and execute this program over the network as if it were making a local subroutine call.

3-2.1.2   The IRIS-PC1-PC2 Network Environment

The prototype virtual workstation emulator network consists of an 80386 personal computer (PC1), an 80286 personal computer (PC2), and a Silicon Graphics Personal IRIS workstation (IRIS). The CPU speed of the Model 4D/35TG IRIS is 35MHz. The IRIS-PC1-PC2 network architecture is shown in Figure 3-1.

```
        ┌───────┐ ┌───────┐ ┌─────────┐      ┌────────┐
        │SPEECH │ │VMKEY  │ │PRINTER  │      │Master  │
        └───────┘ └───────┘ └─────────┘      └────────┘
┌──────────┐    ┌──────────────┐         ┌──────────┐
│  IRIS    │    │    PC1       │         │   PC2    │
│(DISPLAY) │    │  WINDOWS.0   │         │  MS-DOS  │
│  Unix    │    │   /MS-DOS    │         │          │
└──────────┘    └──────────────┘         └──────────┘

        Ethernet (NFS/RPC)
```
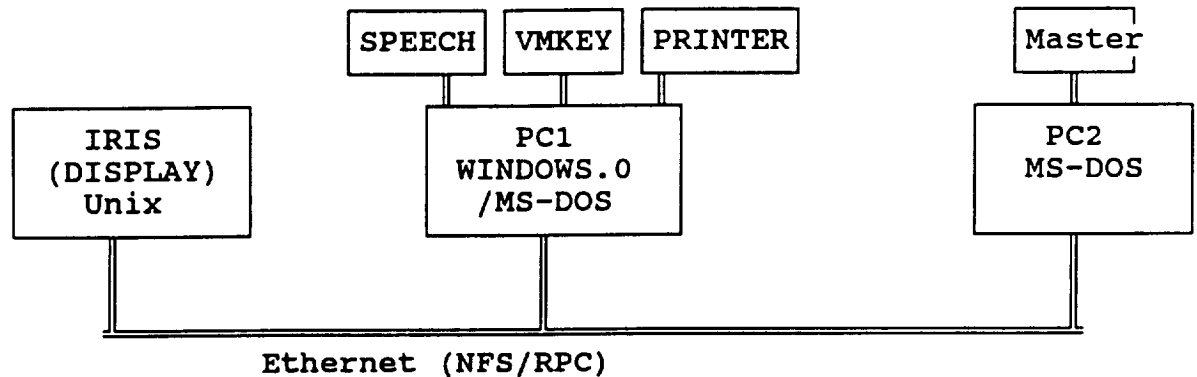
Figure 3-1   Overview of the Prototype Emulator Environment

As shown in Figure 3-1, PC1 was dedicated to the intelligent control and voice command input experiments, and PC2 was used for acquiring and processing data from the fingers/hand/arm position sensing input devices. The IRIS was used for the graphic generation of the virtual scenario elements to be manipulated. The entire system was networked together using Ethernet. This prototype emulator can perform the following experiments:

1) virtual tele-robot arm/hands display and control;
2) knowledge based voice recognition and command input;
3) force-based master arm and position-based master command input and control;
4) intelligent control and decision support system simulation.

3-2.1.3 Communication Option Considerations

1) RS-232

The first option we considered was the simple, low speed serial links between PCs and the IRIS. It is easy to implement and needs almost no extra hardware. The major disadvantage, however, was its bandwidth limitations. RS-232 has a very low bandwidth and does not meet our simulation bandwidth requirement. Another serious problem was the system expansion capability for the Full-Function Emulator FFE) development.

It was decided, after analysis, that the RS-232 be employed in the prototype emulator for data input, operator interface, and connections for system monitoring

nd diagnostics.

2) VME BUS

The second option was a VME bus between PCs and the IRIS. The main advantage is that the resources can be conveniently shared. Its bus drivers are supported by the UNIX operating system so it can be used for communications between different devices. It is particularly useful for large amounts of high speed data transfer and communication, such as high resolution graphics. Also, the VME bus can support data exchanges among the following applications:

a) High bandwidth input data from cameras;
b) High speed computation units;
c) High speed output interface;
d) Small Computer System Interface (SCSI).

The disadvantage of using a VME bus is that it needs a customized hardware and software platform which requires tremendous software development efforts for implementing resource management and sharing. Another problem would be the distance limitation, which could be a serious problem in the near future. VME bus drivers are usually supplied with a Unix system and that can be used to communicate between different devices. A small computer system interface (SCSI) can be interfaced to the VME bus to handle all the data storage, retrieval and development programs.

3) Ethernet Link

The third option, which was our choice, was to employ Ethernet to network the system. The advantages of using Ethernet Link in our experiment were:

a) High speed;
b) Ease of expansion;
c) Availability of software for resource sharing;
d) Availability of cost effective hardware for connecting the systems to ethernet.

Ethernet has a bandwidth of 10 MB/sec. Considering network overhead, the useful bandwidth for the prototype emulator was estimated to be about 300 KB/sec. This has been determined to be sufficient for the prototype emulator.

For inter-process communication over the Ethernet, the "Sockets" protocol was chosen for the ADCACS application. RPC (Remote Procedure Call) and NFS

(Network File System) were also evaluated as options to Sockets.

Internet Domain Stream Sockets was chosen for its simplicity in satisfying the ADCACS project needs. With proper programming, all subsystems can process data concurrently. Any subsystem may be configured as a client or a server. This increases the flexibility of the entire emulator. Minimal overhead was required since Sockets is a fairly low level communications technique.

3-2.2   Graphics

A description of Graphics experiment is included in Chapter Four.

3-2.3   Master Command Input Device

This section involves proprietary information of Astronautics. See Chapter Nine for details.

3-2.4   Knowledge-Based System Control

The PC1 was dedicated for experimental investigations on knowledge-based system control and the decision support system.

3-2.4.1   Knowledge-Based Expert System Control on Voice Recognition

A PC-based speech recognition system, Voice Master Key, manufactured by Convox, Inc., was successfully installed at the beginning of the ADCACS project. The PC1 was used for knowledge-based expert system control of the speech recognition system.

3-2.4.2   A Prototype Control System

Based on PC1 and communication facilities through LAN, the commercial expert system shell, called NEXPERT OBJECT, was used for the following intelligent control experiments:

1)   A Prototype KBS Control System (see Figure 3-2 for details)

2) A simple control system that was based on the time-sequential control strategy. This experiment usee several procedures but implements by one process in PC1. The "calling out" function of the expert system was used, which referred to the ability of an expert system application to call user developed functions or library routines. The NEXPERT OBJECT was employed to build production rules for the data validity checking and control. Data transformation and inter-procedure communication was implemented under the NEXPERT OBJECT shell.
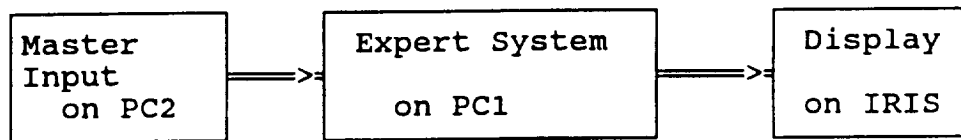
Figure 3-2a: Control of Data Flow from Master Controller to IRIS
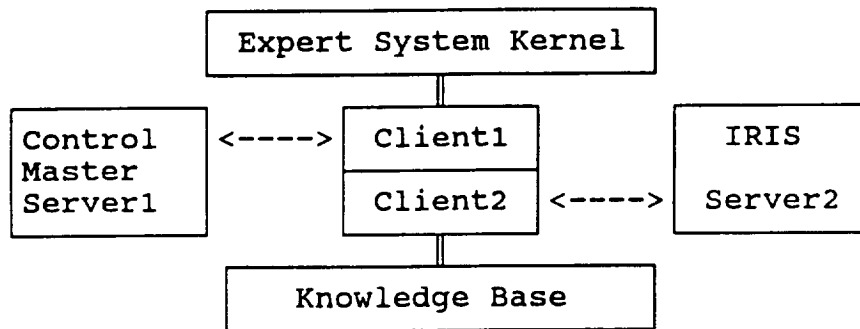
Figure 3-2b: Control Signal Diagram
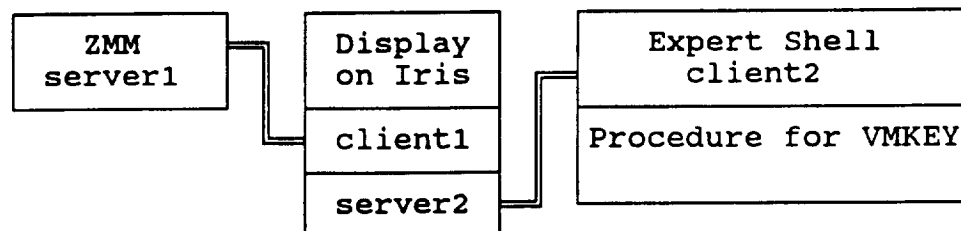
Figure 3-2c: Display Synchronization Diagram

3-2.5   Voice Command Input Devices

3-2.5.1   The Voice Master Key (VMKey)

VMKey Version 2.0 is the voice control device made by CONVOX INC., which can be used to train PC computers to accept commands in simple spoken English -- or any other language. VMK comes in two parts: the hardware "ear" and the software "brain". The "ear" is a combination headset earphone/microphone, plus a plug-in card. This device allows the user to run the application program with voice commands, and thus do more designing and less typing.

3-2.5.2   Voice Command Process Control

The following voice command input functions have been implemented during the experimental investigations:

1) Using VMKEY to accept the control interface signal;
2) Integrating VMKEY into the Expert System;
3) Using SPEECH to demonstrate the voice warning function.

3-2.6   Limitations of the Prototype Emulator

Due to time, information, and budget constraints, the currently developed emulator is only a prototype and has the following limitations:

1) It does not have the Wide-Field-Of-View (WFOV) and LEEP optical devices integrated into the emulating system;
2) The stereoscopic display has not been implemented;
3) The operational warning and caution system has not been implemented.

However, this prototype emulator does establish a solid foundation for construction of the full function emulator which will lead to more intensive technical investigations of the proposed BP/VCWS.

3-3.0   ACT Laboratory Experimental Investigations

The experimental design for the Marquette University Advanced Control Technology laboratory investigation was part of research Activities 6 and 7. The three parts were designed to progressively test and improve the operation of both force- and position-based arm/hand master controller devices (see Chapter Nine for details), to improve the

graphics interpretation of the initial virtual workstation design, and to begin to prove the advantages that a virtual workstation has over the fixed console designs.

It is our desire that the laboratory investigations should be continued in Phase Two if continued funding may be arranged.

3-3.1 Laboratory Experiment Part One (accomplished)

1) Development of a Data Collection Routine

A program was written which was running simultaneously with the experiment performed. This program collected data that were used later in analysis of the results of experiments. The pertinent variables included, but were not limited to:

a) the values read from the analog-to-digital converter for each strain gauge;

b) the specific angles from the force-based master interface processor which were sent across the network;

c) the occasion of a collision detected with the graphics arm to any object portrayed graphically;

d) the location of the collision or object with which the collision occurred;

e) the time in which the data was collected; specifically, the elapsed time between detected collisions.

This was the data necessary to perform and analyze the results of the following experiments, but as these experiments evolved, the data collection were extended.

2) Initial Control Experiment

This first step in the experimentation process involved testing the coarse control of the graphical arm with the implementation of both force- and position-based arm/hand master controller devices (see Chapter Nine for details). The graphical screen was divided into four quarters with each quarter representing a large touch screen. The quarters were assigned numbers 1 through 4. A test subject was placed in the force- and position-based arm/hand master controller and informed of the assigned numbers of the different touch screens. The test began with the experimenter requesting the subject to manipulate the graphical arm to touch the

screen in a specific order. Different sequences were requested and the subject proceeded to satisfy the requirements. The data collection routine described above collected the time and occurrence of the graphical collisions in addition to the order in which each screen was touched.

This experiment demonstrated that collision detection worked well and a coarse level of positioning accuracy was available.

3-3.2   Laboratory Experiment Part Two (accomplished)

This part of the experiments utilized the same four screens as defined in Section 3-3.1, but the four touch screens were displayed at the same time. The screens were displayed in a random order for a predetermined amount of time (e.g. 3 seconds). The subject then was instructed to manipulate the graphical arm to touch the screens while they were displayed. As the subject became comfortable with the manipulation, the time that the screens were displayed will be decreased. This test allowed for an initial degree of analysis of the stimulus - response time, which were enhanced by the arm/hand master.

3-3.3   Laboratory Experiment Part Three (accomplished)

In this experiment, the four screens were incrementally decreased in size. Experiments I and II were modified into one experiment which required more accurate manipulation of the graphical arm. This portion of the experimentation would allow for greater analysis of the stimulus - response characteristics, and provide projected evidence to prove the advantages of the virtual workstation over the fixed console designs.

3-3.4   Laboratory Experiment Part Four (to be performed)

This part was designed for additional virtual display experiments to be performed in Phase Two research. There are 4 planned steps for this experiment:

1)   Two View CRT Presentation.

Two views of a stick figure graphical arm are displayed on a PC-driven CRT. The user must look at both views and mentally fuse the information to be able to operate the force-based arm. This display presentation was chosen for its simplicity so that the force-based arm may be tested. This work has been completed as specified in

the first quarterly report.

2) Single Perspective Presentation.

A single perspective presentation is displayed on a CRT. The display shows a fully rendered graphical arm and hand assembly. A graphical arm and hand have been completed. The lighting model has been chosen to simulate space by using one infinite light source and little ambient light. The lighting model is variable and could include several local lights mounted on a FTS.

Construction of the hand and arm has been influenced by the rendering speed and required quality of presentation. The virtual elements need to be rendered within 1/30 of a second for interactive real-time presentation. Computational time is being reduced by creating an image requiring as few polygons as necessary. The number of polygons will be reduced as long as the time necessary for the operator to identify and appropriately react to the presentation is not reduced. The determination of which polygons are necessary includes consideration of the visual elements used to identify the object. Since the orientation of the virtual elements will alter the time necessary to render the presentation, the worst case rendering time is used.

3) Stereoscopic Presentation.

Liquid Crystal Shutter (LCS) glasses are synchronized to a 120 Hz CRT. Each eye sees a different perspective view causing the operator to perceive a three dimensional image. Although the directional positioning of the eyes is correct, the focal length does not change. The effects on the operator caused by these uncorrelated positioning cues needs to be explored.

One problem is the potential for one eye seeing a ghost of the image created for the other eye. This can occur for two reasons. First, the persistence of the phosphor may cause a ghost of one image to remain on the CRT while the other eye is viewing the present image. Second, the LCS may let small amounts of light through while it is closed.

While the LCS is open, the transmissivity may be less than 100% causing more power to be consumed for a given brightness.

The LCS glasses stereo viewing option has been installed on the Personal IRIS. The LCS is used in the third step of display experimentation because the technology is immediately available and is relatively inexpensive.

4) Stationary WFOV.

Of the two optical designs, the chosen one has several advantages. These advantages include: no ghosting problems; no LCS transmissivity problems; LCS synchronization and driver electronics are not necessary; a standard 60 Hz CRT and not a 120 Hz CRT is needed. If a WFOV is determined not to be necessary for the needed application, similar and less expensive optics may be used in a similar arrangement.

Whether or not WFOV is necessary needs to be determined. It is possible that in space peripheral vision is not needed to operate a FTS. This determination will consider the tasks that the FTS needs to perform and the lighting available or needed for the tasks. Such lighting models are available in this experimental setup.

3-4.0   Computer Hardware and Software Study for the Proposed BP/VCWS Design

3-4.1   Micro-Processor Technologies

The technology of high performance and parallel processing microprocessors is growing very rapidly. It is desirable that the most advanced microcomputer be used as the BP/HPC system central processor. Following are several candidates for the BP/HPC central processor:

1) The Intel 50 MHz i860 RISC Microprocessor

This microprocessor represents the latest microprocessing technology in the field. It is a 64-bit CMOS microprocessor that contains more than 1 million transistors. The chip's core is a RISC (Reduced-Instruction-Set-Computing) processor. In addition to standard operations, i860 can, like a supercomputer, perform vector operations. The i860's overall performance is for most applications about half the speed of Cray-1. Because the i860 has the same data storage structure as the 386, operating systems for the 386 can be easily adapted to the new i860 environment. For software developers, Intel has a tool kit (assemblers, simulators, debuggers, etc.) and a C language compiler.

2) The Intel 50MHz 80486 Microprocessor

This processor is the newest member of Intel 80x86 microprocessor family. It is 80386 compatible, with enhanced high speed and superior performance.

The Intel i860 will be the ideal candidate for both the BP/VCWS central processor

and the graphics engine/image processor. It should be designed, however, to be compatible in software, interfacing, and module assembly with other processor ORUs. The replacement of the i860-based processing unit should be designed in such a way that it could be replaced by a 386 processor ORUs, which still will be operating, but only with downgraded performance.

3-4.2 Software Development Considerations for the Proposed BP/VCWS

According to the BP/VCWS design and performance goals, the software design and development procedure should achieve the following objectives:

1) low development and maintenance cost;
2) assurance of the functional correctness;
3) flexibility and simplicity for the future upgrade and modification.

3-4.2.1 Operating System (OS) Environment Overview

The Body-Ported High Performance Controller (BP/HPC) is the central computer for the BP/VCWS. It will use the operating system kernel that is compatible with that of the SSF DMS system, with the enhancement of functions that provide special cupola operational control functions. The BP/HPC system central processor operating system will have the following elements:

1) OS kernel;
2) System and utility software (device drivers, file manager, etc.);
3) OS shell and development tools.

The Operating system will work under the multi-tasking and DMS networking environment. It also will provide inter-communication and coordination between different task processes.

The BP/HPC system central processor operating system will have the capacity to handle and incorporate the following software programs required by the cupola operations:

1) Knowledge-based control and decision support software;
2) Graphical and "window-like" user interface software;
3) Real-Time Graphics Display and Image Rendering Software;
4) Conventional control system software;

5) Data processing and file management software;
6) Communication, networking, and interfacing software;
7) Maintenance software.

3-4.2.2 Knowledge-Based Control and Decision Support Software

_ The role of using knowledge-based, or expert system software for intelligent Mission control and decision support systems is specified and in accordance with NASA SSF Advanced Automation Program. The following functions, which are a subset of cupola functions and operations, specified in Chapter 2, will be implemented and controlled by the knowledge-based system:

1) EVA and other operational monitoring in the Space Environment;
2) Caution and Warning Systems (C&WS);
3) BP/VCWS DMS System Testing and Diagnosis;
4) Other emergency handling.

Knowledge-based system control and decision support software will be developed by employing the NASA Expert System CLIPS tool, which is developed by NASA JSC. CLIPS is the abbreviation for "C Language Integrated Production System". It has superior advantages over existing commercial expert system shells. A detailed description is included in Chapter Five.

3-4.2.3 Graphical User Interface Software

The BP/HPC software will provide a multi-media user interface that supports multi-window graphics, hypertext display, and voice/master command input and interaction.

3-4.2.4 Real-Time Virtual Display and Image Rendering Software

The software will support the "virtual reality" display functions of the BP/VCWS. The designated software will be able to operate real time.

The image rendering software subroutine will operate under the main display software program that executes the low-level graphics/image processing and rendering. This is implemented on the graphics engine with high performance RAM and local storage to meet the high speed graphics image rendering requirement. The image rendering graphics engine will have a bandwidth capacity of 100MB/second.

3-4.2.5 KBS Control Incorporation with Conventional Control

The following cupola control function will be implemented by the conventional control (as opposed to the KBS control) software in cooperation with the KBS control software:

- Teleoperated and telerobotic systems control.

Also, the conventional control software will provide functions for the training and simulations.

3-4.2.6 Data Processing and File Management Software

This software will provide general services to the crew operator, such as data processing, document retrieving, word processing, and file management.

3-4.2.7 Communication, Networking, and Interfacing Software

This software will provide interfacing and networking between the BP/VCWS and the DMS network. It will also provide communication functions between the cupola crew operator and other crew onboard the SSF. Specific functions include electronic mail service, audio/video communications, and conferring.

3-4.2.8 Maintenance Software

Maintenance software is a part of the system house-keeping software which will include diagnostics routine, computer virus protection routine, etc.

3-4.2.9 Entertainment Software

Entertainment software contains various computer game routines for the entertainment of cupola members during their spare time. This is only an optional item.

3-4.2.10   Software Development Automation

1) Cost Reduction of Software Life Cycle

The automation of the BP/VCWS/HPC software development is motivated by potential benefits, such as reduced software life cycle costs.

–   Software life cycle costs, including development and maintenance costs, have been increasing rapidly due to various factors, such as growing software functional complexity and increasing labor costs. It is necessary and practical to consider applying new design methodology and using automated software development tools to reduce the elapsed time between the beginning of software planning and the time of delivery.

The objectives stated above can be assured by following software engineering methodology, and by employing state-of-the-art software automation technology. The standard software development procedure will be the "waterfall" software life cycle model, which can be described as a procedure:

    a) Software system development feasibility, specifications, and validation;
    b) Software plans and requirements validation;
    c) High level software structured design and preliminary review process;
    d) Detailed design and verification;
    e) Coding and testing;
    f) Software integration and product verification;
    g) Software system test;
    h) Operations, maintenance, and upgrading.

2) Proposed Approach for Automating BP/VCWS Software Development:

    a) Use of NASA COSTMODL software development cost analysis and estimation tool.

    b) Use of commercially available CASE (Computer Aided Software Engineering) tools.

    The CASE (Computer-Aided Software Engineering) tools will be employed for the purpose of reducing the cost. Both Upper CASE, which refers to the automated high level tools for structured software design, and the Lower CASE, which in turn refers to the automated coding tool, will be used.

c) Use of Objected-Oriented Design and Programming (OOD and OOP).

Though both C++ and Ada can be applied to accomplish this goal, the C++ is preferred to Ada for the use of BP/VCWS software development. Moreover, it has the capability of interfacing with programs written in Ada, which is one of the proposed languages for the SSF DMS system.

d) Knowledge Acquisition Automation

One of the major concerns in software development is the knowledge acquisition process, which performs knowledge transfer from human experts to the expert system knowledge-based software. The proposed knowledge acquisition procedure for the development of the BP/HPC knowledge-based system will follow an automated "repertory-grid" methodology, proposed by John Boose at Boeing Company. See Chapter Five for details.

CHAPTER FOUR

STUDIES ON VIRTUAL DISPLAY, OPTICS, AND GRAPHICS RENDERING TECHNOLOGY

4-1.0   Organization of Chapter Four

This chapter consists of the following items:

4-2.0   Display Technologies;
4-3.0   Optical System Devices;
4-4.0   A Proposed Laboratory Experiment Plan for Hardware Solution to the LEEP Display Prewarping Problem
4-5.0   Laboratory Experiments on Virtual World Generation and Graphics.

4-2.0   Display Technologies

4-2.1   General Analysis

A thorough evaluation of evolutionary display technology requires the evaluation of many different performance parameters. However, a limited number of performance measures and evaluation criteria establish whether or not a display technology is a suitable candidate for the specific BP/VCWS application. These considerations include resolution, chromatic display adequacy measured in several terms, physical size, weight, power consumption, and ease of control.

When evaluated in terms of the evolutionary technical requirements of the BP/VCWS, the present observation is that the current state-of-the-art helmet or head-mounted display systems do not meet the needs of the proposed BP/VCWS. The main shortcomings are in the areas of resolution and high fidelity color display. The conclusion, which is based on the current research, is that specific application requirements may be met with lower resolution displays, and that at least 1000 x 1000 lines will be required for BP/VCWS. The issue of color presentation is constrained by the number of pixels of the display, screen updating frequency, and the bandwidth of the graphics computation and rendering.

1) Liquid Crystal Display Technology (LCD)

LCD is light in weight, and its size can fit easily in the head-mounted assembly. Currently, it is employed by the NASA/Ames virtual workstation. However, the current available resolution is limited to 500 lines. Another disadvantage of LCD systems is their low optical transmission capabilities which requires back illumination systems that consume significant amounts of power.

2) Cathode Ray Tube (CRT) Technology

In CRT technology, monochrome video display tubes with geometries suitable for body-ported or head mounting (i.e., diameters in the range of 1-2 inches), can be obtained with 1000 lines resolution. At the present time, the state-of-the-art in this size of tube is limited to monochrome display. It is predicted, however, that the 1" diameter, 1000 line, beam index CRT will be commercially available within the next 24 months. This new device will certainly be a feasible candidate for installation in the BP/head-mounted assembly.

3) Distributed Cathode Display Technology

A promising technology that appears able to meet the requirements of the BP/VCWS design is the flat-cold cathode CRT utilizing a distributed cathode technique. This technology offers the following advantages:

a) High Resolution -- up to 300 full color lines per inch. This means that a 1000 x 1000 line color display could be provided in a 3.3 x 3.3 inch panel which could be ideal for integration into head-mounted assembly;
b) Reduced power when compared to LCD displays;
c) High MTBF;
d) Matrix addressable that is compatible with digital image generating systems;
e) High light output;
f) High frame and field rates.

It is definitely one of the best choices for the BP/VCWS display system. However, schedule risk exists for the time needed for testing and installation. it is predicted that it will not be available until 1993. It is anticipated that the distributed cathode display will be the first candidate for display technology (in the near future) when it becomes technologically mature and commercially available.

4-2.2   Display System Evolutionary Design

The need for much higher resolution than can be found in a Head-mounted display is not driven by a specific task goal, but rather is driven by the need for maximum flexibility in evaluating the benefits and drawbacks of the higher resolution system. Some of these experiments will determine the value of increased resolution on task performance. Others will evaluate the effect of the Wide-Field-Of-View (WFOV) in conjunction with stereoscopic images in a high resolution environment. The flexibility of this design approach will allow a greater scope in experimental investigation.

The proposed evolutionary display system design is as follows:

1)   Perform the on-ground emulator display design and experiment by using Silicon Graphics IRIS Workstation which can provide the desired resolution and stereoscopic requirements.

2)   Apply the CRT display devices that are being employed by the US Air Force "Super-Cockpit" project for the Head-Mounted Display Design;

3)   Continue looking for commercially available display products in the near future to replace existing designs until a satisfactory design is achieved. One of the promising devices is recently announced by CAE Electronics of Canada. Further investigations remains to be done.

4-2.3   Display Design Alternatives for the Full Function Emulator

The following two alternative design approaches for display of the Full Function Emulator have been evaluated. These design approaches are shown in Figure 4-1 and Figure 4-2. The design in Figure 4-1 uses a single monitor with two liquid crystal shutters over the screen. The liquid crystal shutters would present left and right eye views at 120 frames per second (60 Hz per eye). The shutters would be synced to the graphics engine which would alternately draw the left and right eye views, with each eye image using approximately half the screen pixels. These images would then be viewed through the LEEP viewer. This necessitates the compression of the image into the LEEP format before it is displayed.

The second design choice in Figure 4-2 uses two separate monitors for the alternate right and left eye views. This removes the need for any kind of shuttering system as well as giving each eye the entire pixel count of the screen at 60 times per second per eye. The LEEP format is required for the display as in the previous approach.

The choice that is most promising is the two monitor approach. It is conceptually simpler with better display quality. This is in line with our earlier statement that we do not want to limit the range of the experimentation any more than necessary by our choice of hardware.

The LEEP format is not presently available as a standard option on the Personal IRIS. It may be possible to render the virtual elements into a LEEP format using software and by-passing the graphics engine. This would require more processing time to render the picture and real-time manipulation would probably not be possible. Yet optic and human factor experiments could possibly be performed to analyze the LEEP viewing format.

4-3   Optical Devices

4-3.1   Optical System Design Goals

NASA's desire for a more efficient and productive BP/VCWS design for the Space Station Freedom necessitates a thorough examination of all optical design choices. In order to fully evaluate the most effective design choices in the design of the workstation, it is necessary to start with the maximum possible quality that can be obtained and then determine where these criterion can be relaxed without sacrificing the overall design goal. For this reason, the optical system designs that have been considered at the Marquette University ACT laboratory have all been of the peering type. The justification for this design approach is technological limitations.

The current display technology does not give the kind of pixel density required for a head mounted wide-field-of-view stereoscopic display device. This was shown by Fisher at NASA Ames with the head mounted display developed at Ames. The LEEP compression lens elements, developed by Eric Howlett at Pop-Optix Labs and used in the Ames HMD, require much greater pixel density than was available at that time to provide a good quality display. The rapid advances in flat panel display technology should overcome this limitation in the next few years.

The need for much higher resolution than can be found in a head mounted display is not driven by a specific task goal, but rather, is driven by the need for maximum flexibility in evaluating the benefits and drawbacks of the higher resolution system. Some of these experiments will determine the value of increased resolution on task performance. Others will evaluate the effect of the Wide-Field-Of-View in conjunction with stereoscopic images in a high resolution environment. The flexibility of this design approach will allow a greater scope in the experimental investigations.

4-3.2   Laboratory Optical Experiment Design

The choice that the ACT lab has made is the LEEP stereoscopic viewer optics that are produced by Eric Howlett of POP-OPTICS LABS. His viewing optics are made from six large glass elements which allow one to get a 90 degree direct field-of-view and a total Field-Of-View of 140 degrees when the corneal field is taken into account. This larger total Field-Of-View results from refraction at the surface of the cornea. This is a very important aspect of realistic immersion in the scene.

4-3.3   Engineering Considerations

Some engineering problems should be considered as the optical system develops:

1) The LEEP optics suffer from chromatic distortions. These can be compensated for by varying the size of the different color images. The main problem with this is that it will add to the computational overhead of the system which slows recalculation of the images;

2) The prewarping required in order to obtain high quality central image is another computational burden again slowing recalculation of images;

3) The overall system potentially has several input devices operating concurrently, also adding to the computational overhead;

4) Potential solutions to part of the computational burden:

   a) The greatest computational burden is the additional graphical computations necessary to restore the correct perspective and color to an image viewed through the LEEP optics. This problem could be solved by designing a dedicated hardware device, which is a component of the graphics engine, to perform the transformation (see Section 4-4.0 for detailed analysis);

   b) A second option is the use of a specialized graphics engine with the ability to do faster image recalculation and also the ability to perform scaling (See Chapter Six for detailed analysis).

4-4.0   A Proposed Hardware Solution to the LEEP Prewarping

The Head Mounted, 3-D Stereoscopic Color Display System is a subsystem of the Advanced Display and Computer Augmented Control Systems (ADCACS). The purpose of this subsystem is to give the user the illusion of being in the virtual world by displaying an image in the user's Wide-Field-Of-View. This is accomplished by displaying the image of a screen viewed by the user through optics. The optics warping, or mapping, the screen image into the user's Wide-Field-Of-View. The screen image must be *prewarped* so when viewed through the optics, a natural, perspectively correct, Wide-Field-Of-View image is presented to the user.

Software generation of a prewarped image from an unwarped image causes significant computational overhead, reducing image recalculation speed. This problem could be solved by designing dedicated hardware devices to preform the prewarping transformation. The solution proposed here is achieved by preforming the prewarping using analog circuitry in the CRT display. The prewarping of the display image is accomplished in the CRT display using analog circuitry by preforming the prewarping transformation on the vertical and horizontal plate voltages.

This proposal represents a practical solution to a complex problem, allowing concentration of resources on other issues in the Head Mounted, 3-D Stereoscopic Color Display System while the higher speed, higher resolution prewarping transformation solution is under development.

The potential advantages of the proposed prewarping are:

1)   no computational overhead.
2)   simple circuitry, easily modified to match optics.
3)   can be implemented quickly, allowing other aspects of the research to continue

However, the disadvantages may be:

1)   at best, does not improve image pixel resolution;
2)   cannot interpolate between image pixels in the vertical direction of the CRT screen.

4-4.1   Analysis of Prewarping

The prewarping of the displayed image varies the radius of a particular pixel from the center of the display, while maintaining the angle of the pixel. An example of this is

shown in Figure 4-3. Let (x,y) and (r,ϕ) be the location of an unwarped pixel in cartesian and polar coordinates, respectively, and let (x_w, y_w) and (r_w, ϕ_w) be the location of an prewarped pixel in cartesian and polar coordinates, respectively, where the origin is the center of the display. The prewarping is achieved using the following radius transformation.

$$r_w = F \cdot [r - ar^3]$$
$$\phi_w = \phi$$

(1)

where F is the focal length and a is a function of the index of refraction.

Parameters x_w and y_w are calculated using

$$
\begin{aligned}
x_w &= F \cdot [r - ar^3] \cdot \cos(\phi) \\
&= F \cdot [r - ar^3] \cdot x/r \\
&= F \cdot [1 - ar^2] \cdot x \\
\\
y_w &= F \cdot [r - ar^3] \cdot \sin(\phi) \\
&= F \cdot [r - ar^3] \cdot y/r \\
&= F \cdot [1 - ar^2] \cdot y
\end{aligned}
$$

(2)

Figure 4-4 shows the effect of this prewarping transformation on a regular rectangular grid. Near the origin, the warping moves the pixels away from the origin, mapping more of the screen to the center of the field of view. Near the edges, the pixels are moved closer together than near the origin. Figure 4-5 shows a pixel prewarping transformation on a 320 by 200 pixel screen.

4-4.2   Design of the CRT Display

The typical color CRT monitor is illuminated by three electron beams (one for each color) sweeping horizontal lines in a top-to-bottom fashion. The screen consists of a matrix of phosphor dots of three different colors. Directly behind the screen is a shadow mask aligned with the screen such that each electron beam strikes the phosphor dots of only one color. The beam is typically thicker than the phosphor dot pattern on the screen, so that several phosphor dots are illuminated at any given instant in time. For a simplified model, the horizontal and vertical beam deflections from the center of the

screen is proportional to the horizontal and vertical deflection plate voltages. Thus the unwarped plate voltages are saw-tooth waves. The prewarping transformation can be implemented in the CRT monitor by applying the prewarped voltages to the horizontal and vertical deflection plates. If x and y are the unwarped horizontal and vertical plate voltages, then the prewarped horizontal and vertical plate voltages are given in (2).

The prewarped horizontal and vertical plate voltages, x_w and y_w, can be generated with circuitry implementing the block diagram in Figure 4-6.

4-4.3   Implementation of the CRT Display

The simplest way to implement the block diagram in Figure 4-6 is with analog circuitry to modify the horizontal and vertical plate voltage references. The circuit can be implemented with four analog multipliers or logarithmic amplifiers, and operational amplifiers for scaling, multiplication by constants, adders, and limiters. Interpolation between in the horizontal direction can be accomplished by low-pass filtering the x_w signal.

4-4.4   Conclusions

A practical solution to the display prewarping transformation has been proposed. This transformation is achieved by modifying the CRT monitor deflection plate voltages. Although this solution is not proposed as the final solution to the problem, it will allow development of other aspects of the project while the higher speed, higher resolution prewarping transformation solution is under development.

4-5.0   Laboratory Experiments on Virtual World Generation and Graphics

Graphical generation is performed on the IRIS. This machine is fast enough to generate the real-time (no perceivable delay) images necessary for the present research. It is anticipated that as the complexity of the control scenarios increases, the graphical image generation power will have to be increased.

The primary graphics program presently in use is "armbox.c". It uses the Silicon Graphics standard graphics library, GL. GL is a set of C callable subroutines. The subset of routines being used in "armbox.c" includes:

1)   a variety of viewing windows;

2) a single infinite light source (multiple infinite and local lighting source models are available for experimentation and require increased processing time);

3) RGB mode and Gouraud shading for realistic shading with fewer polygons, as compared with flat shading;

4) double buffering to synchronize rendering with the screen so smooth motion exists with no flickering;

5) backface removal and z-buffering to decrease rendering time for hidden surface removal;

6) viewing matrix manipulation and stack routines to easily and quickly position graphical elements;

7) queues for internal events and standard input devices;

8) simple 3 dimensional polygon rendering routines.

The program routine "armbox.c" renders a picture of a right arm, including a hand and fingers using positioning data received over the network. It also has techniques for drawing boxes in any location and orientation. For iteration experimentation of the graphics portion of the emulator, a second process may be run on the Personal IRIS which emulates a microcomputer sending data to the IRIS.

For the networking setup, the IRIS is the client and PC2 is the server. The main graphic loop requests positioning data from the server. It then immediately renders a picture using initial data. After rendering the picture, it reads the data sent from the server and loops. The second time through the loop it again requests data from the server and immediately renders a picture, this time with the previous data. Each time through the loop the picture is rendered using the previous data. This technique allows the graphic generation to be performed in parallel with the acquisition and preprocessing of the positioning data.

The screen is refreshed at 60 Hz so there is no perceivable flicker. The image update time appears to be slower than 60 Hz. Until the next image is generated, the IRIS simply re-displays the same image. For the same 3 dimensional model, the image generation time will differ depending on the depicted position of the virtual arm.

Collision detection has been successfully completed for boxes in any orientation. In addition to drawing an arm, several boxes are also drawn within arm's reach. The location of the tip of each finger is compared to the location of a box resulting in a Boolean array of collision locations. Additional points on the fingers, hand, or arm may be identified as important for collision detection and added to the Boolean array. This may be done for any number of boxes.

The present collision information is used to draw the appropriate box with a color

that is dependent on which finger is touching the box. This gives the operator clear visual feedback.

Figure 4-7 to 4-15 show some examples of the Graphics generation. The C language source code list is included in Appendix A.

4-11

TOP VIEW

SCREEN FOR LEFT AND
RIGHT EYE VIEWS

TWO TEXTRONIX
LIQUID CRYSTAL
SHUTTERS ONE
FOR EACH VIEW

LEEP VIEWER

PERSON VIEWING
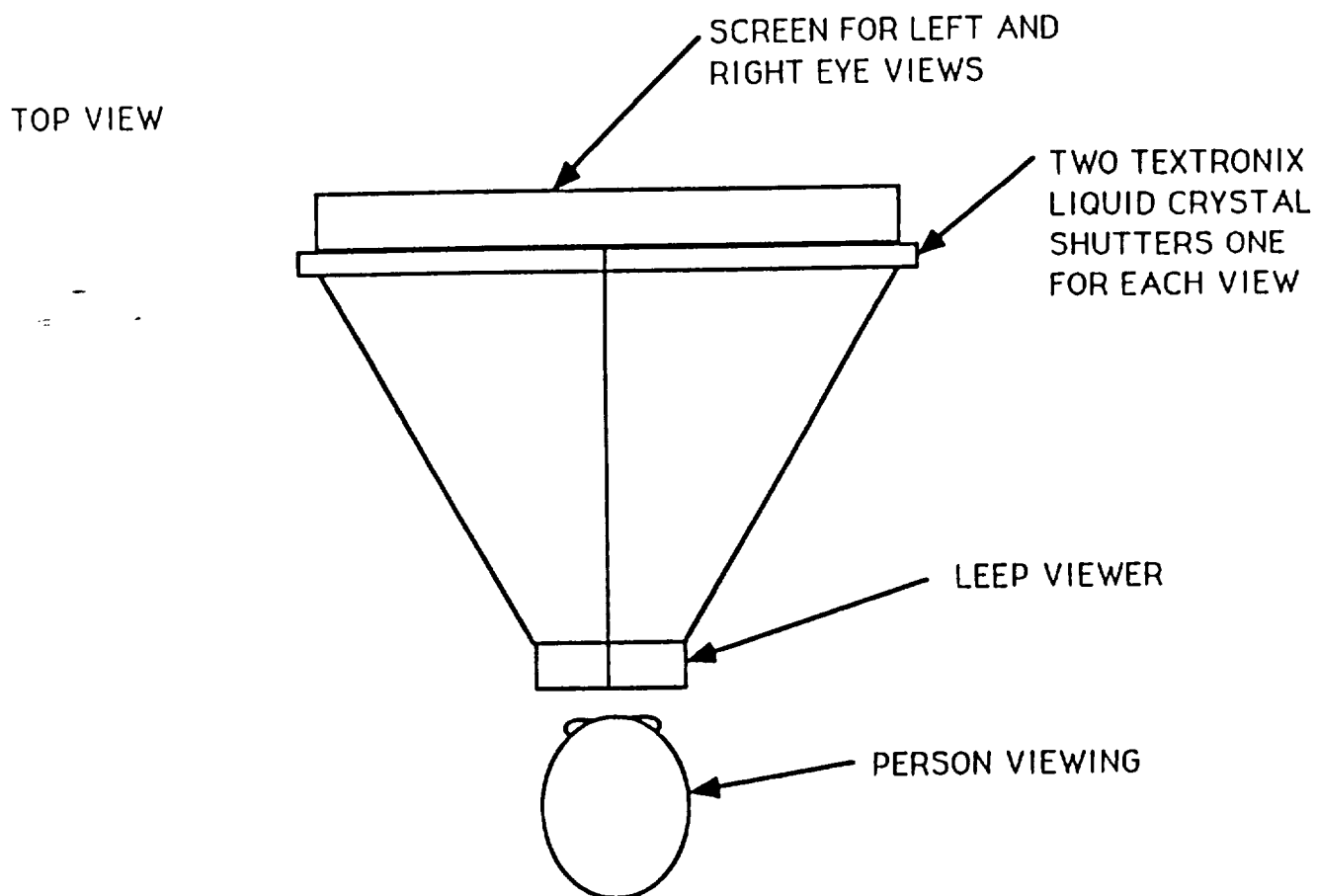
Figure 4-1

Alternate design which was rejected due to a loss of resolution
using this approch.

LEFT EYE SCREEN   RIGHT EYE SCREEN

TOP VIEW

OPTICAL ELEMENTS
NEEDED TO SCALE
IMAGE FOR LEEP
VIEWER.

LEEP VIEWER OPTICS

PERSON VIEWING
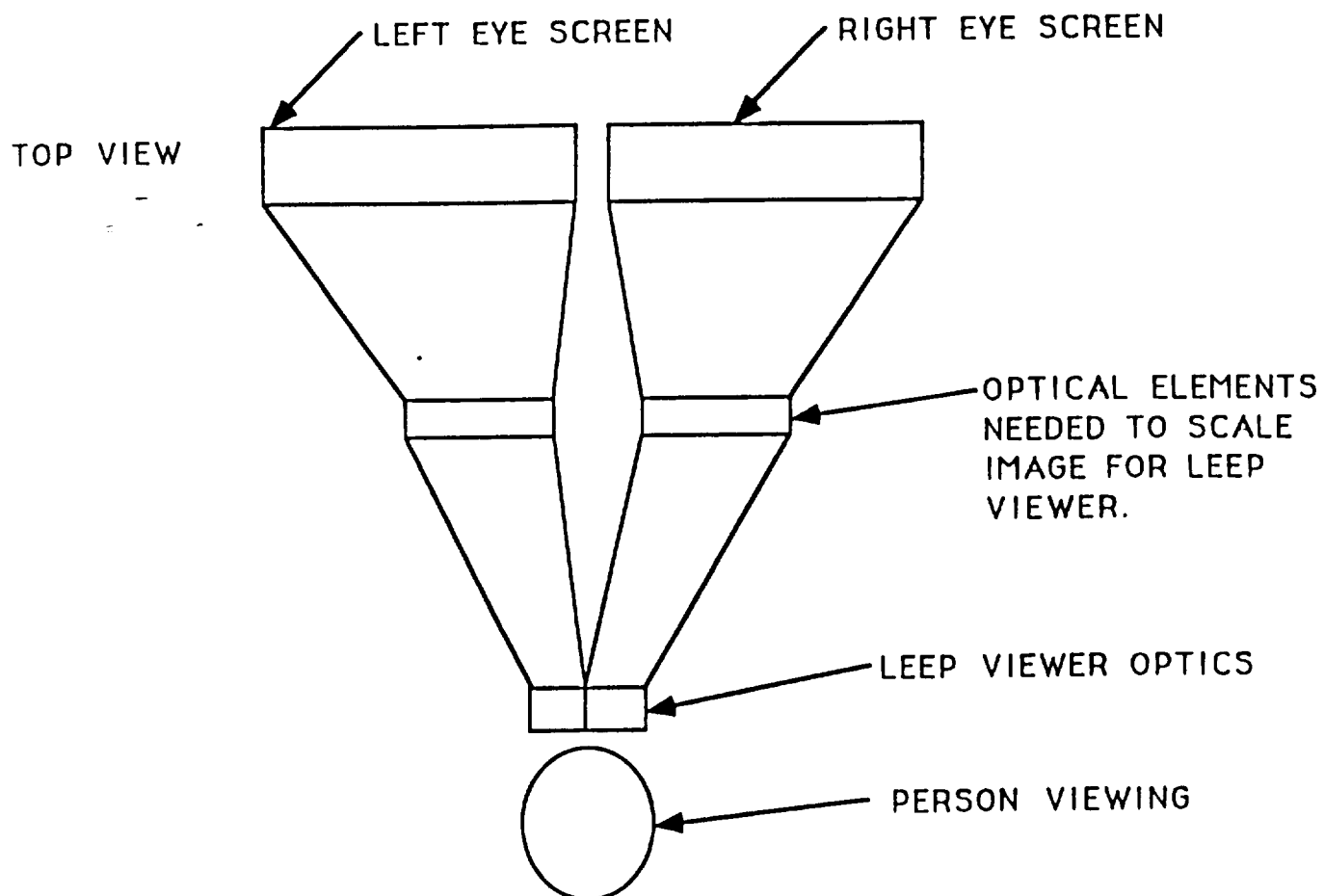
Figure 4-2
Perposed Optical Emulator design using two monitors
  for presenting the alternate right and left eye views.
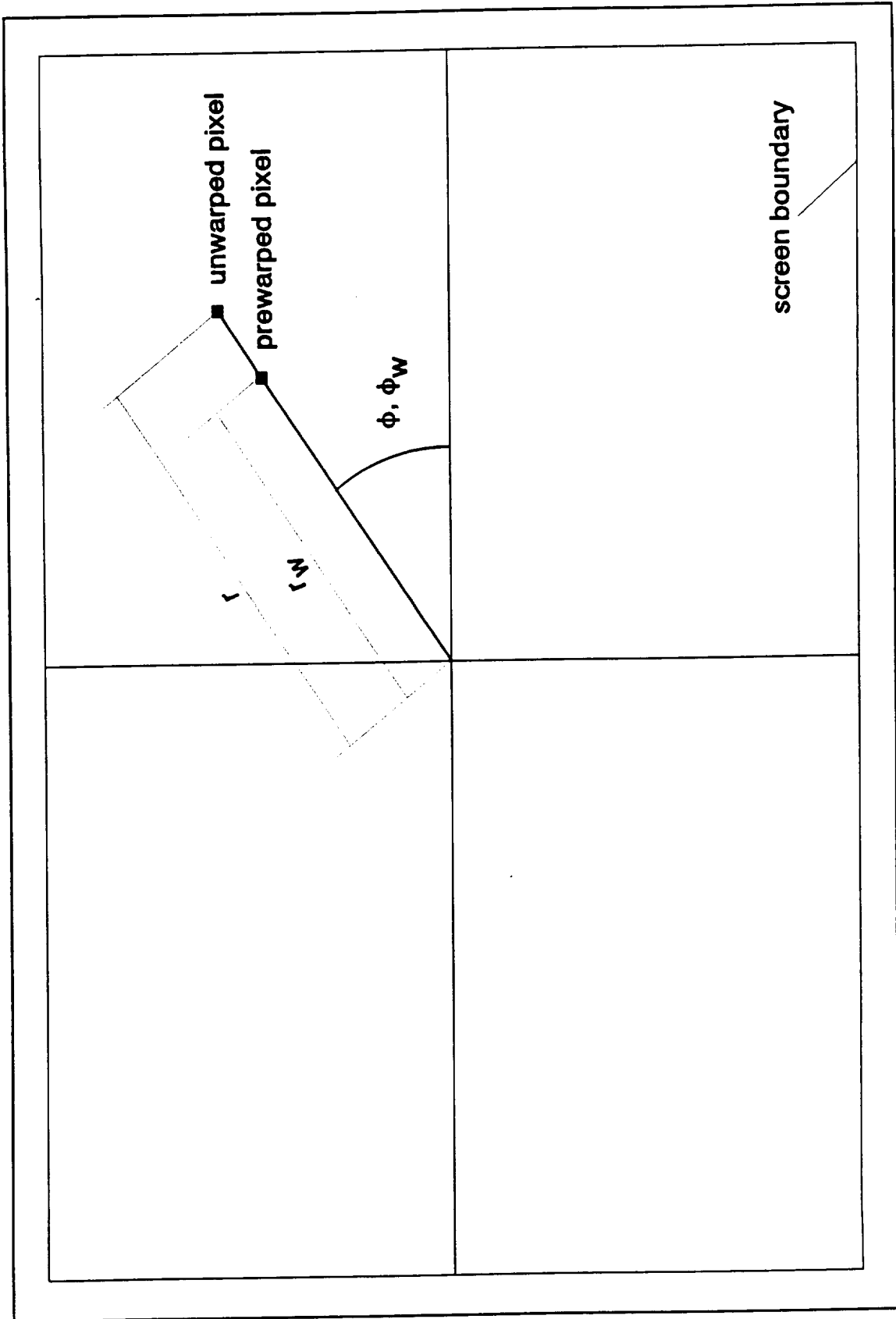
Figure 4-3   An example of a pixel prewarping transformation.

F=1.301 σ=0.12
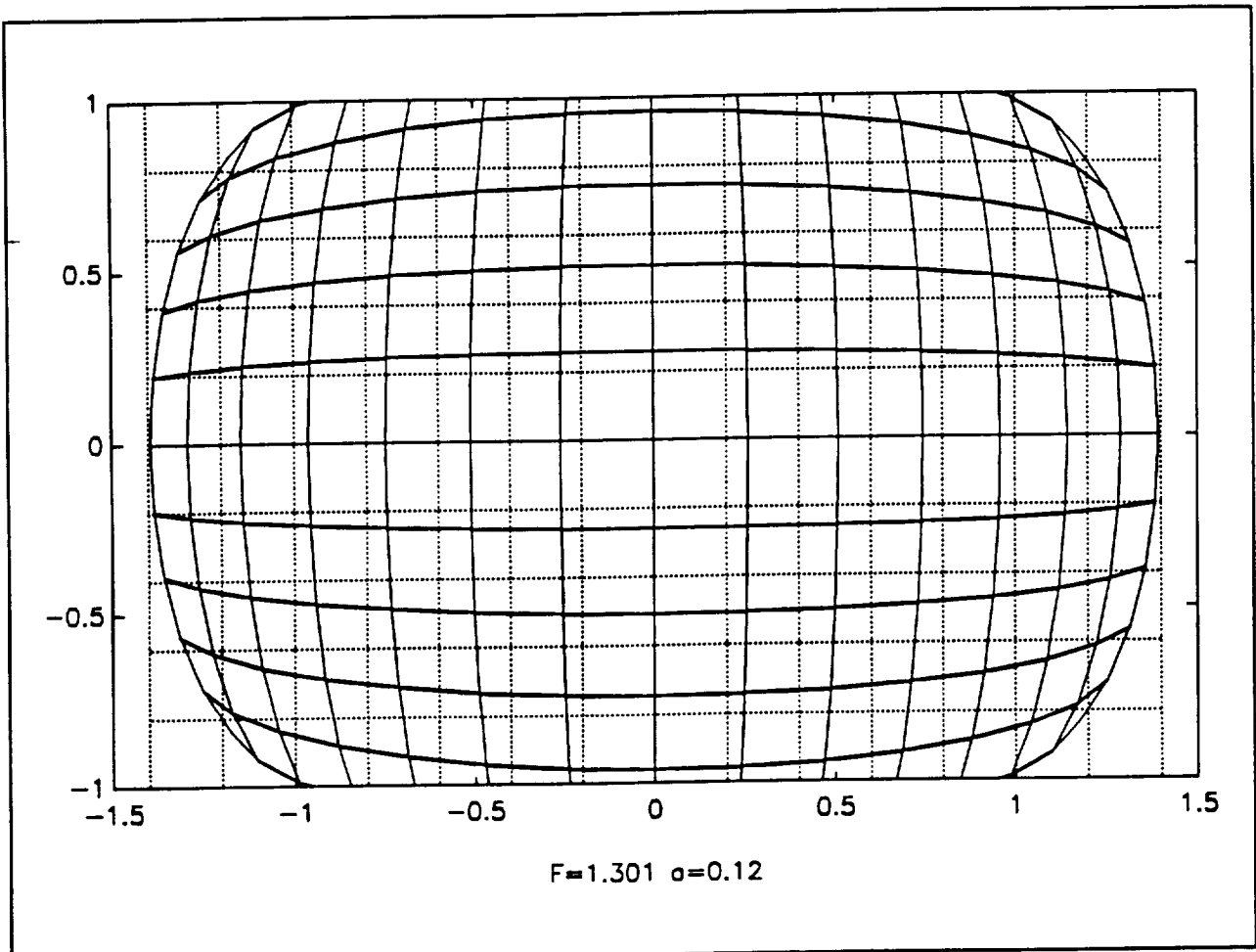
Figure 4-4 Unwarped (dotted) and prewarped (solid) display images.

pixel map at 320 x 200
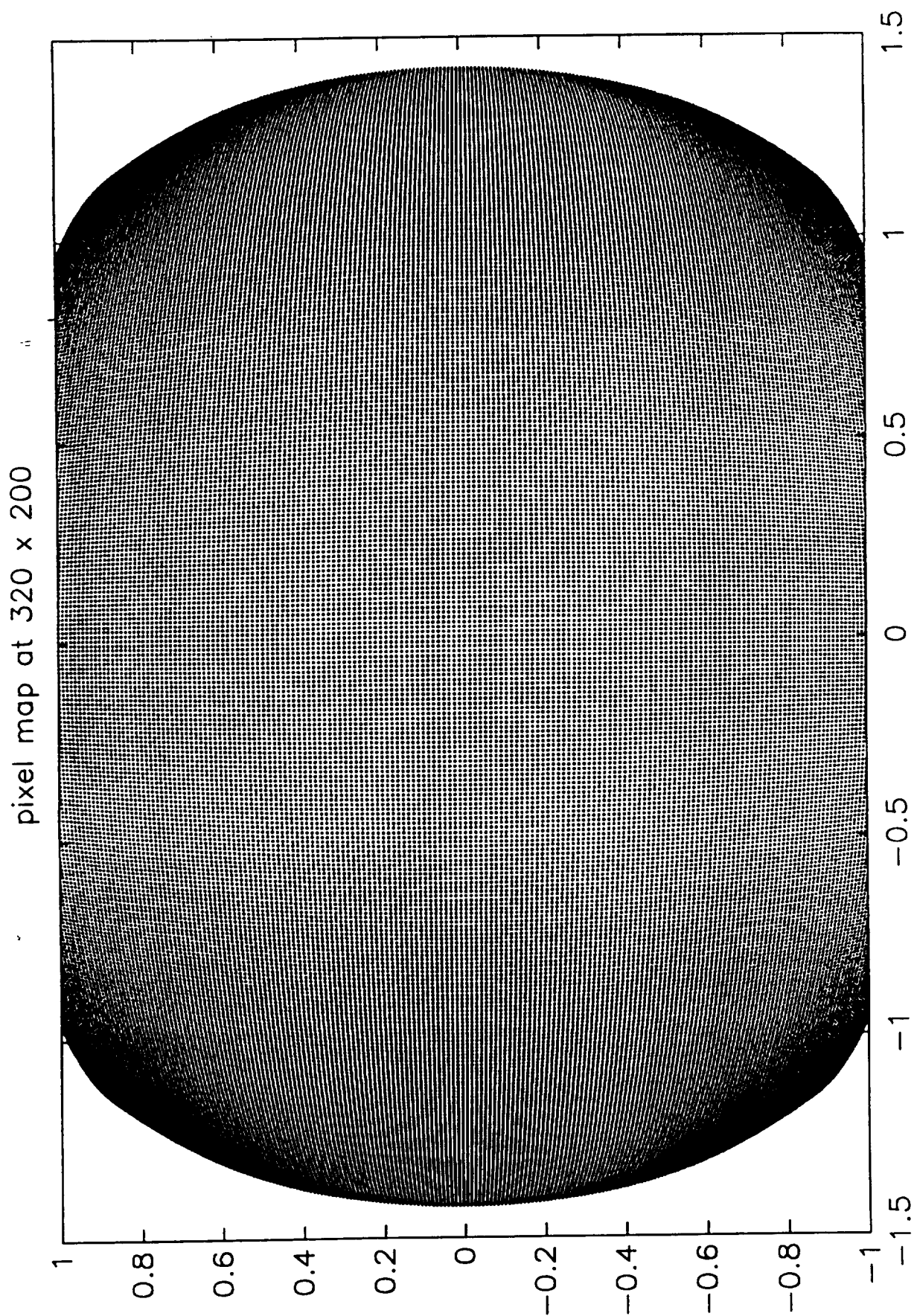
F=1.301 a=0.12

Figure 4-5 Pixel Prewarping Transformation on a 320 x 200 Pixel Screen
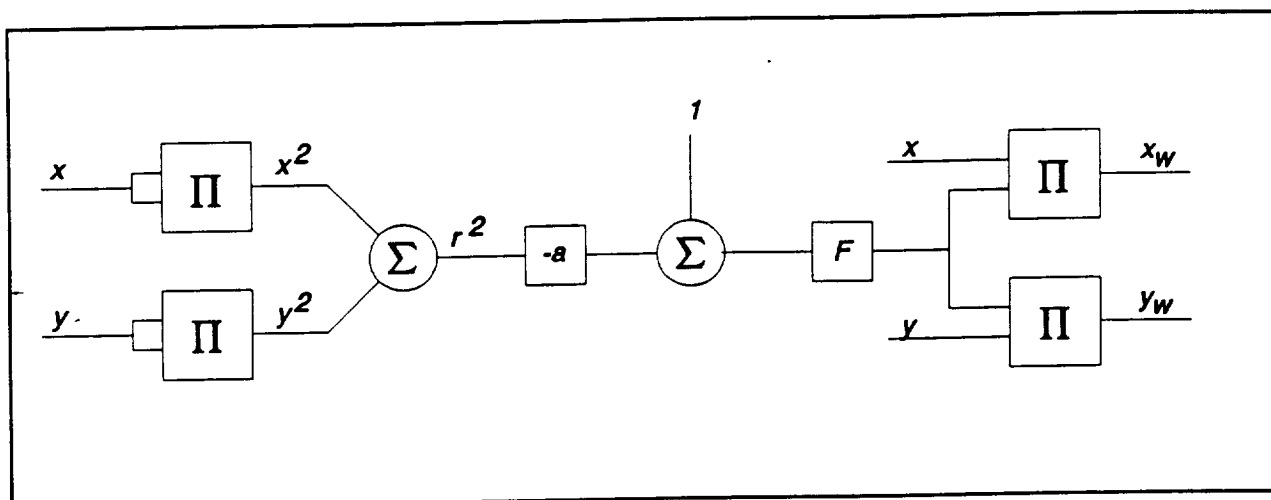
Figure 4-6   The block diagram of the prewarping transformation for the CRT plate voltage.
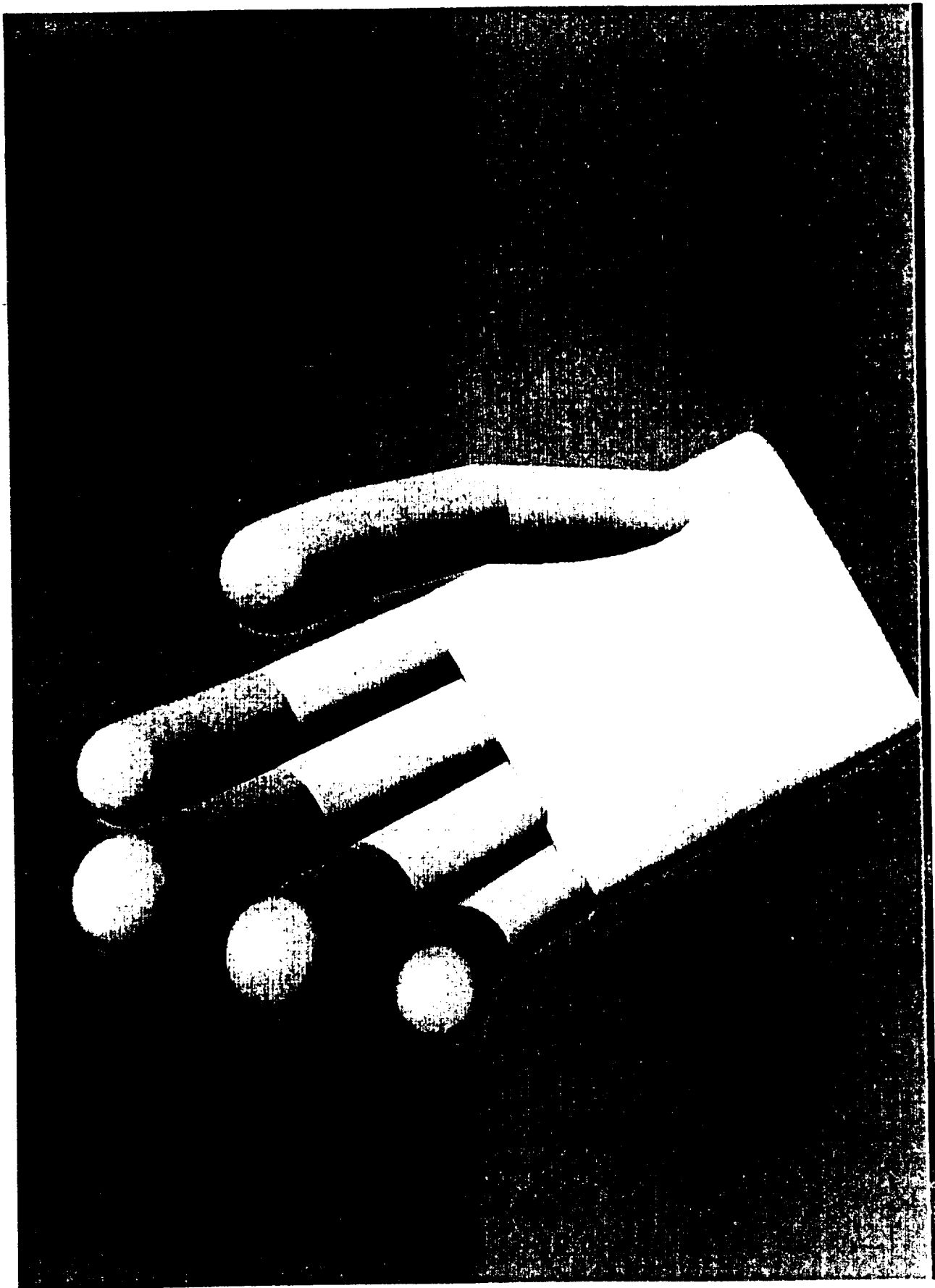
Figure 4-7 Examples of Graphics Generation
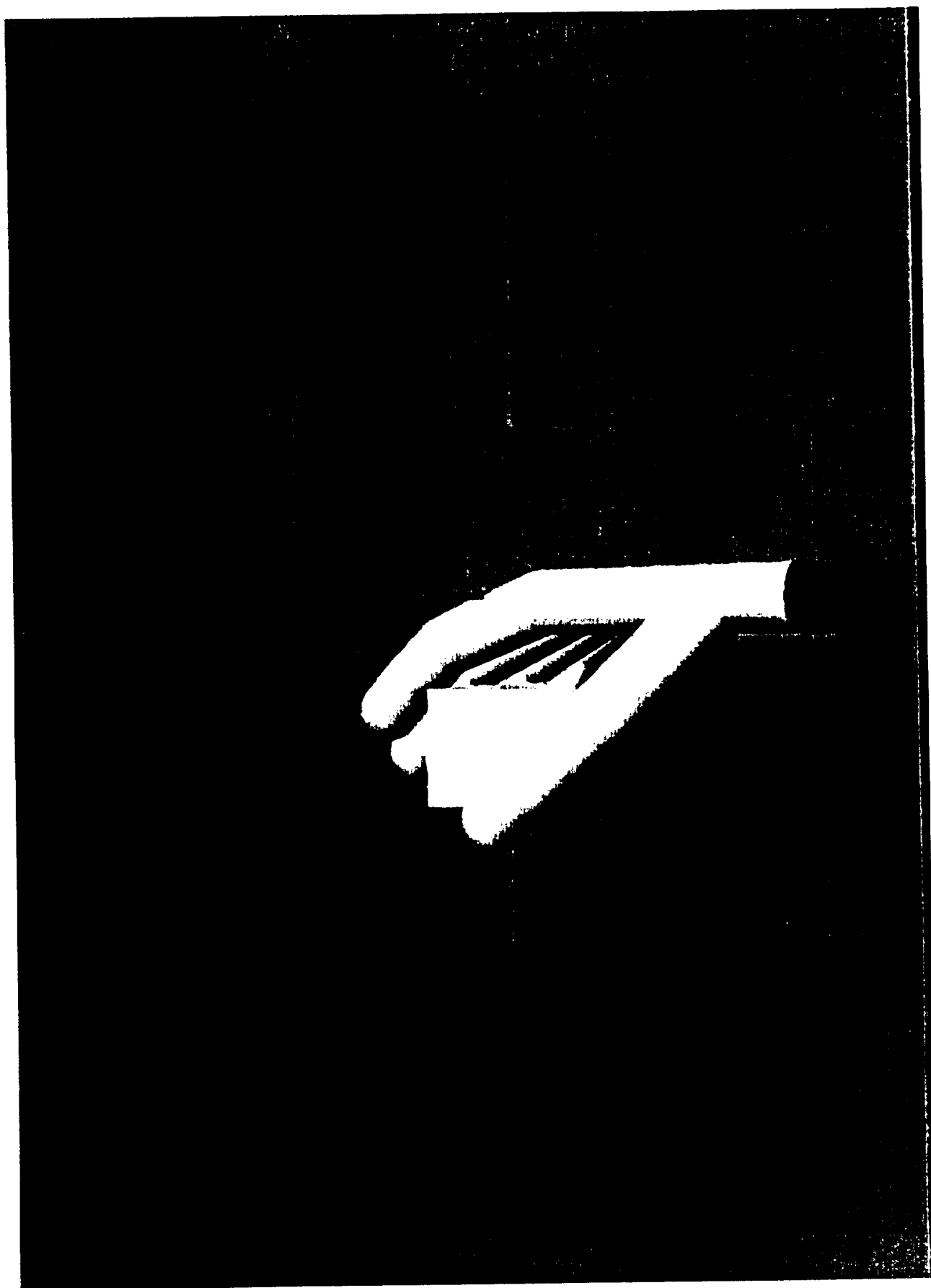
Figure 4-8 Examples of Graphics Generation
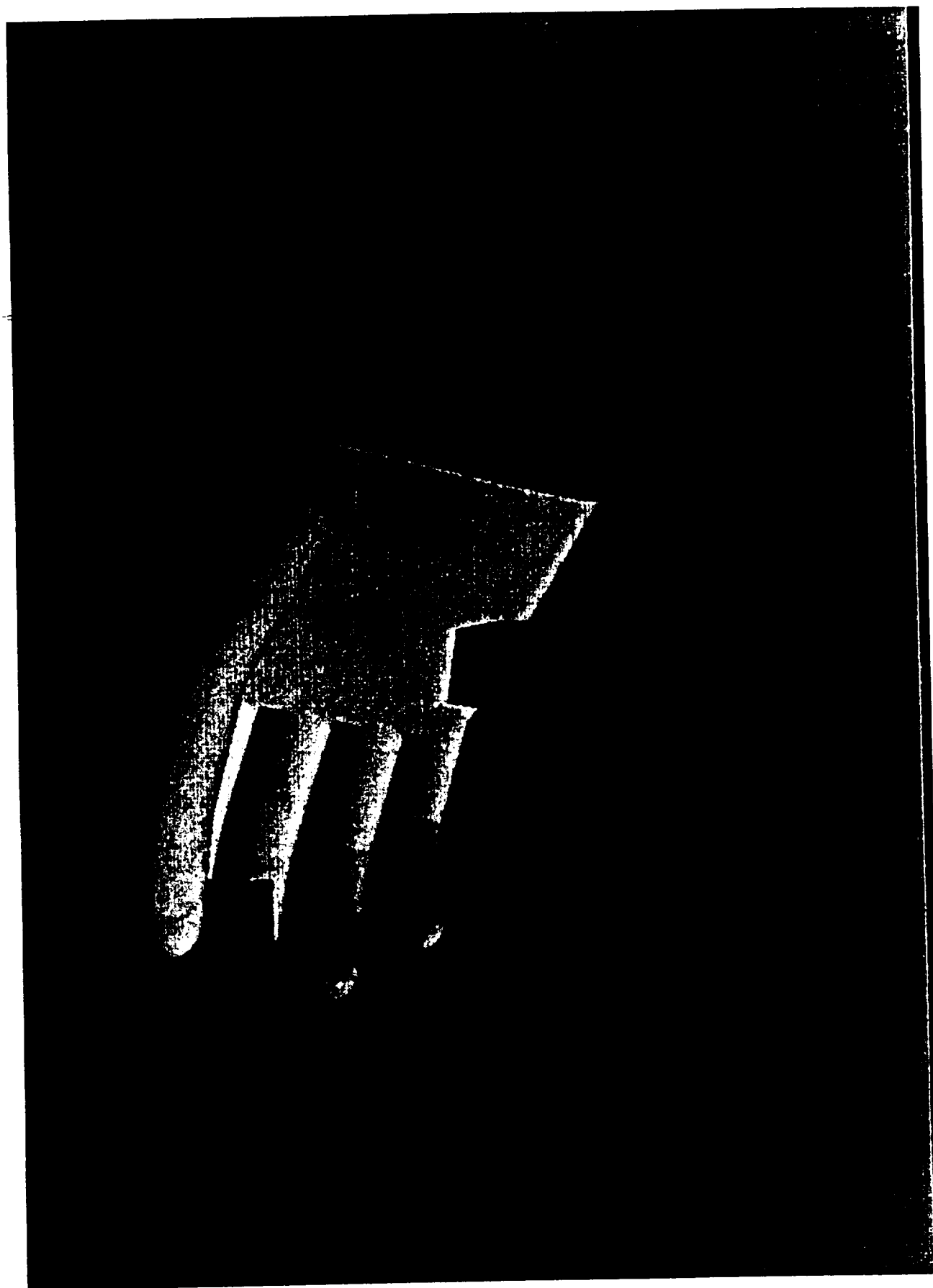
Figure 4-9 Examples of Graphics Generation
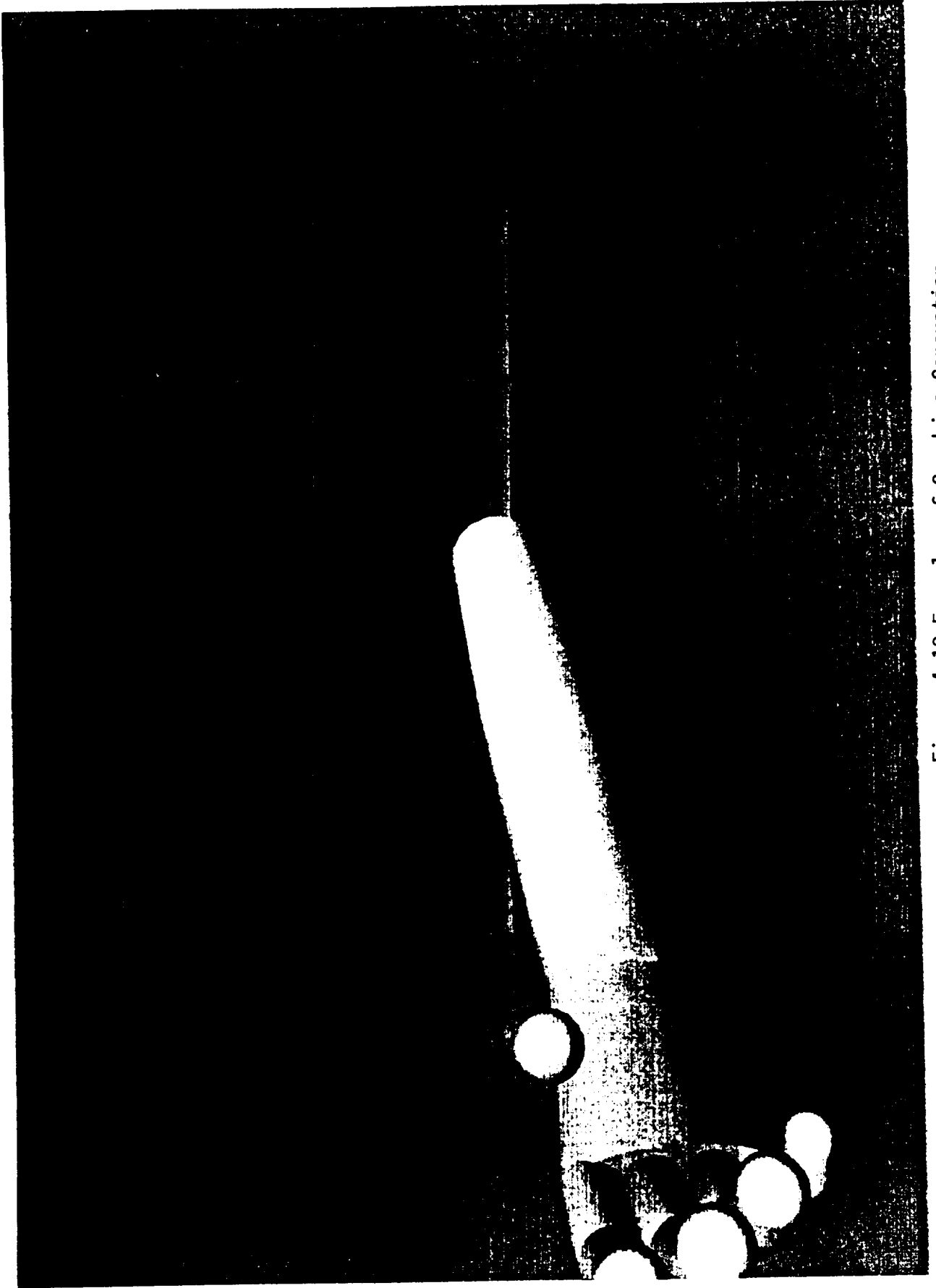
Figure 4-10 Examples of Graphics Generation
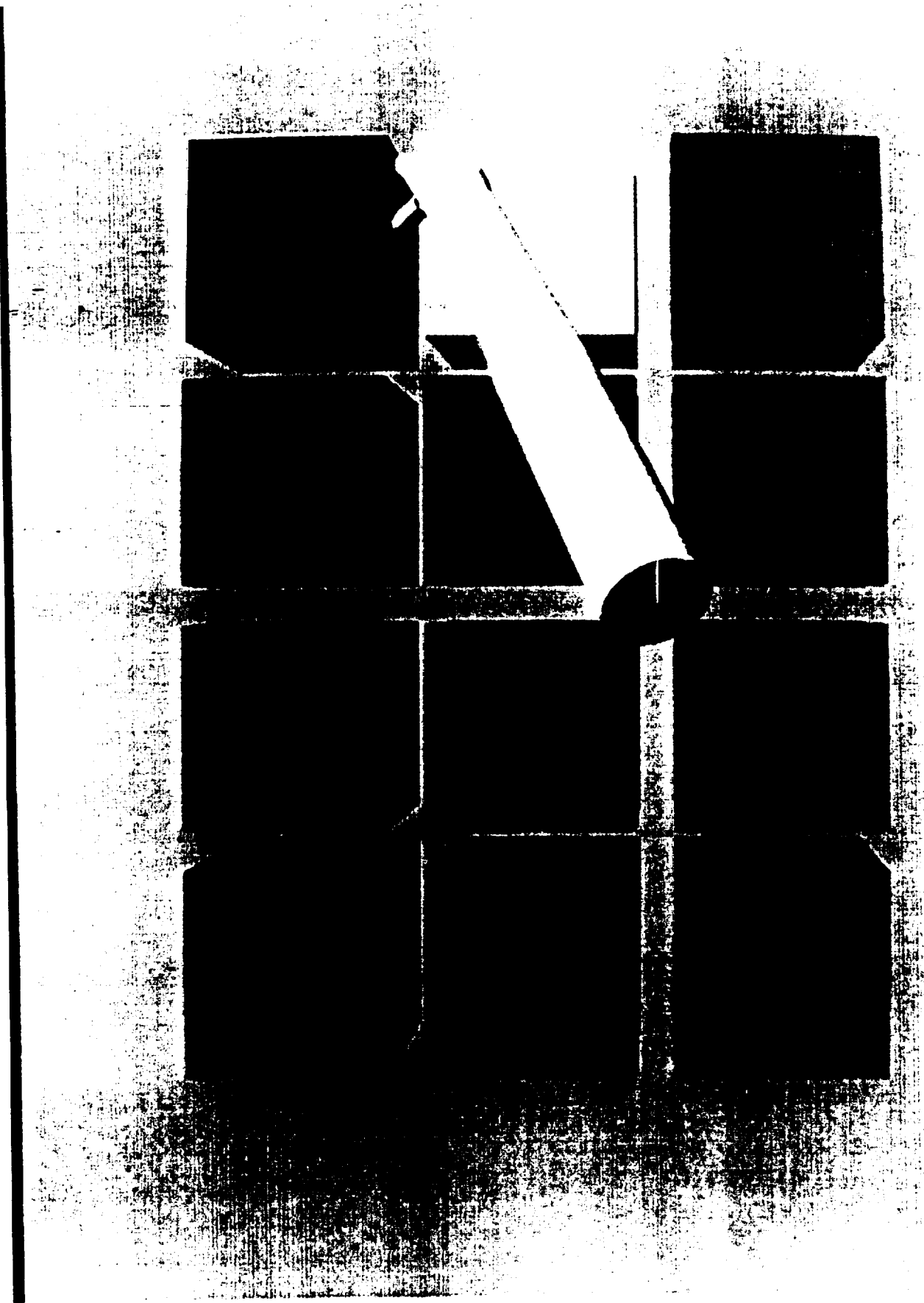
Figure 4-11 Examples of Graphics Generation
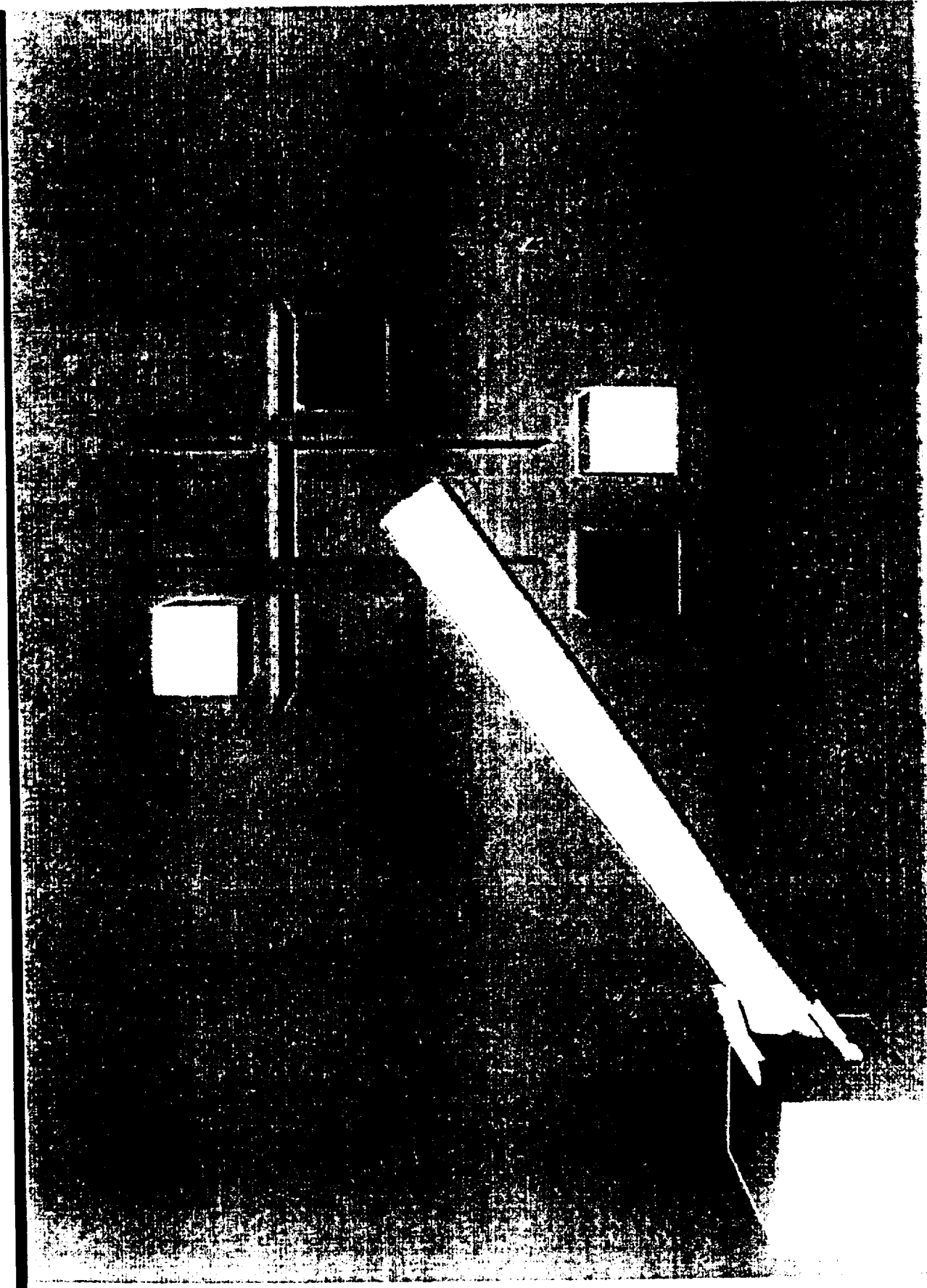
Figure 4-12 Examples of Graphics Generation
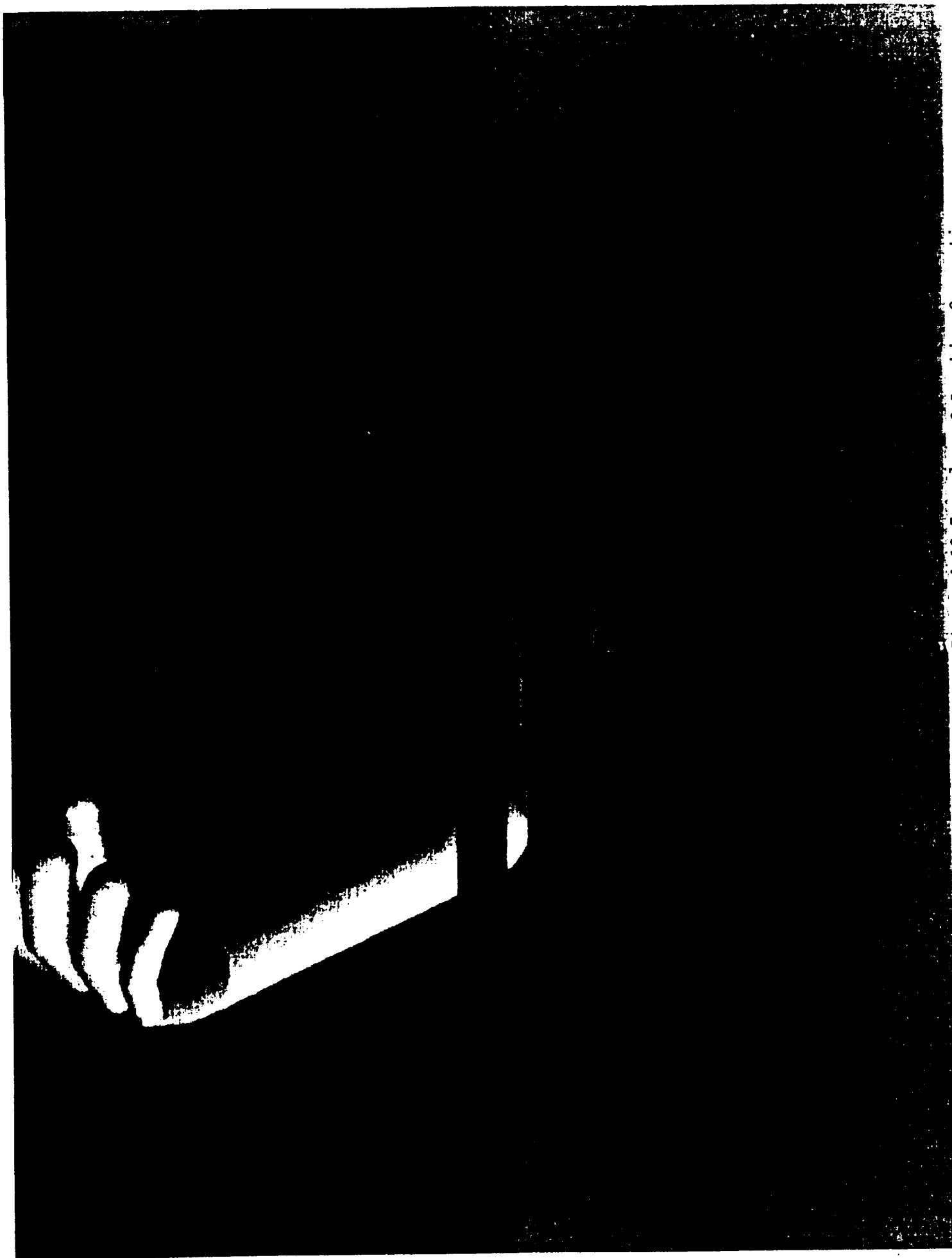
Figure 4-13 Examples of Graphics Generation
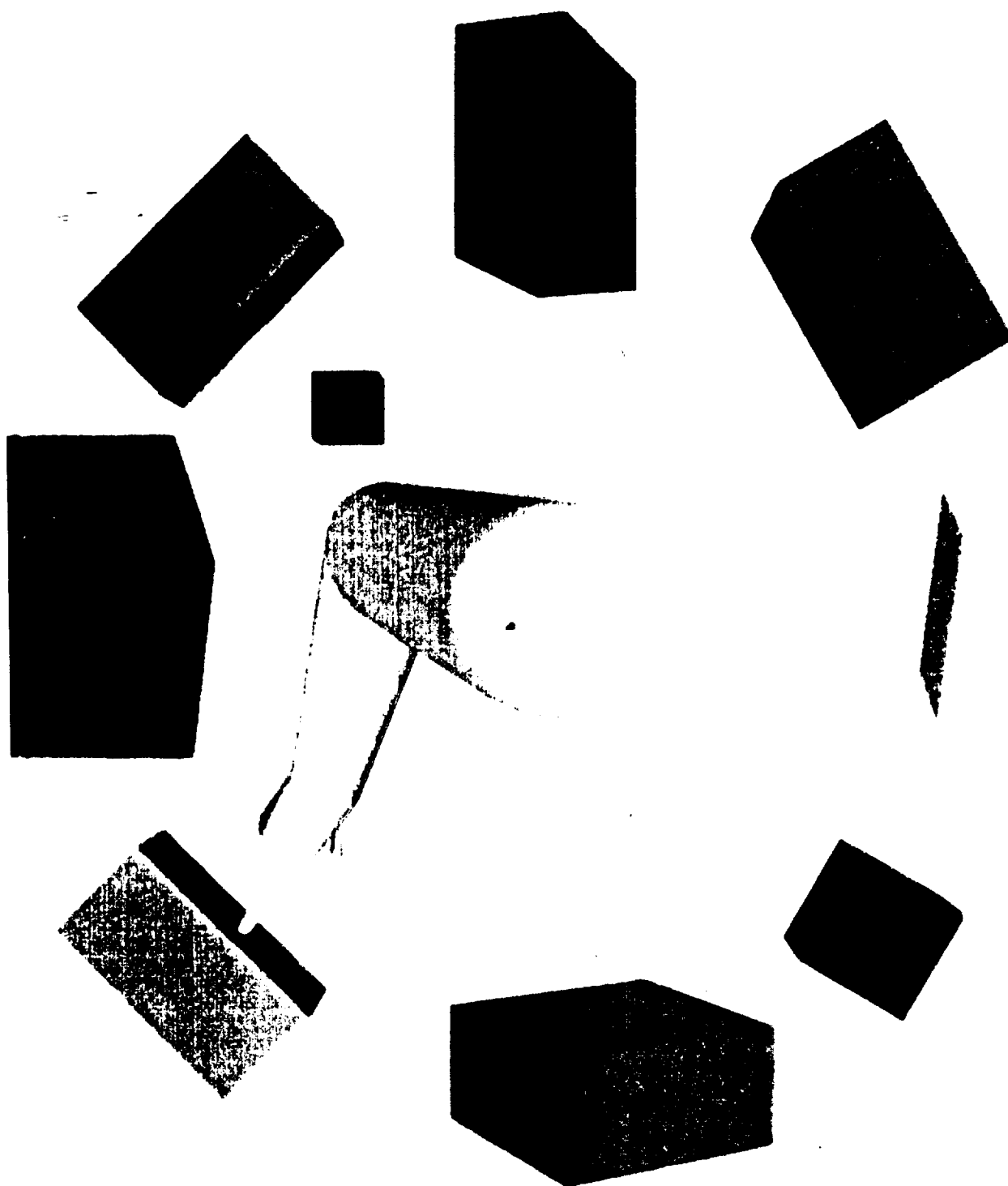
Figure 4-14 Examples of Graphics Generation
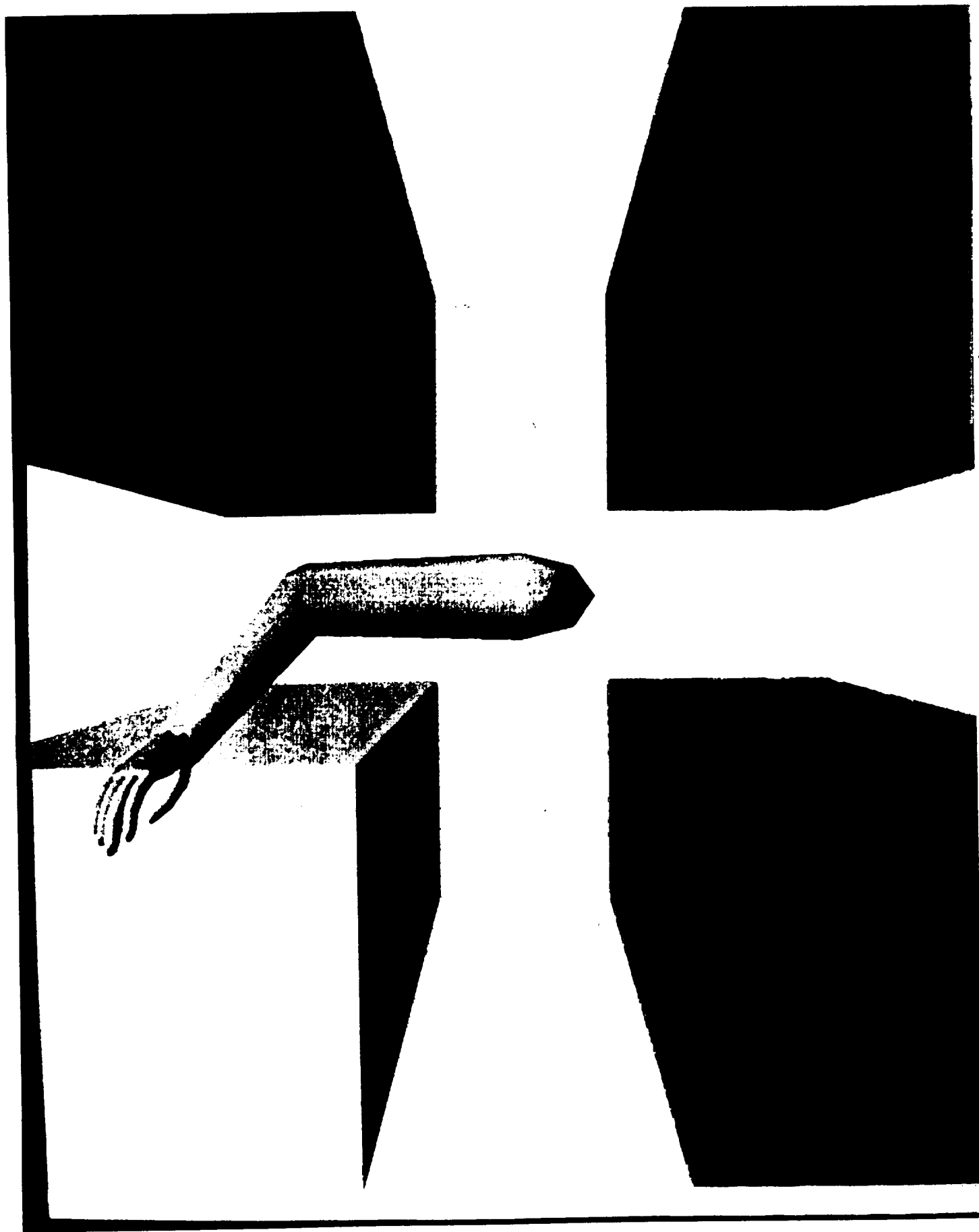
Figure 4-15 Examples of Graphics Generation

CHAPTER FIVE

FEASIBILITY STUDY OF KNOWLEDGE-BASED SYSTEM (KBS) CONTROL FOR THE PROPOSED BP/VCWS

5-1.0   General Role of Knowledge-Based System Control for BP/VCWS

The most significant feature of the software system for the BP/VCWS is that it applies Space Station Advanced Automation Technology (SSF AAT) to provide intelligent Mission control and decision support functions for the BP/VCWS. The Knowledge-Based System will play an important role and will be imbedded into the control and monitoring software system.

5-1.1   Roles and Impact of the SSF Advanced Automation Program

According to the NASA Space Station document: "A Review of Space Station Freedom Program Capabilities for the Development and Application of Advanced Automation," prepared by MITRE Corporation MTR-88D00059 in December, 1988, advanced automation is defined as "the use of concepts and methods of high-level symbolic inference by a computer and the symbolic representation of the knowledge to be used in making inferences so as to make a machine behave in ways that humans recognize as intelligent behavior." This definition was restricted to advanced automation only and, therefore, excludes robotics.

Advanced automation will offer a wide range of potential benefits to the BP/VCWS, from fault detection/isolation, to intelligent mission controls. It is apparent from various studies that future space station designs should increase the mission productivity and safety, and reduce both development and maintenance costs. These concerns and ultimate goals have been reported in several NASA Space Station documents. Studies demonstrate the significance and importance of promoting advanced automation development for early applications of these goals. Since the proposed BP/VCWS employs intelligent and expert system mission control technologies, it is obvious that the BP/VCWS system will play an important role in advanced automation technology. The on-going ADCACS project is certainly moving in the direction of reaching the goals and meeting the future space station cupola operation requirements.

5-2.0   Strategies of Using Knowledge-Based Intelligent Control

As advanced automation technologies mature and standards are defined, reliable and cost-effective systems can be built using these new technologies if the laboratories, test beds, SSE, and MSIF are prepared to support their application. Currently, the Knowledge-Based Systems (KBSs) are considered the most mature of the new technologies in the category of advanced automation. No longer do KBSs correspond in an approximate way to those performed in a conventional software development project, although they are performed in a somewhat different approach. To support these differences, existing development and test facilities need not be replaced; they need merely evolve to support these new technologies.

Two development environments must exist within the SSF and BP/VCWS to promote advanced automation technology to the required level of technology readiness: 1) environments for state-of-the-art technology development; and, 2) state-of-the-practice application development. For the SSF program, technology development is nurtured in laboratories and test beds while application development is supported in the SSE. Evolution of BP/VCWS in the SSFP is linked both to the laboratories and test beds, as well as to the SSE software factory. These existing facilities should evolve technologically in parallel with the evolution of the Space Station Freedom.

The following technical issues relating to the existing facilities have been identified and are determined in this report.

5-2.1   Determination of KBS Development Tools

The NASA developed CLIPS (C Language Integrated Production System) expert system development tool has been identified, tested, and compared with existing commercial KBS tools. The recommendation has been made that it will be used for KBS development of both the full function BP/VCWS Emulator, and the final deliverable KBS intelligent mission controller for the BP/VCWS.

1)   Advantages of Using CLIPS

CLIPS is an open structure expert system tool. It is written in standard ANSI C, with enhancements of user-friendly window interface and object-oriented design capacity. Because of its portability, extensibility, capabilities, low cost, and especially the availability of C language source code, CLIPS has received widespread acceptance throughout government, industry, and academia. The development of CLIPS has helped to improve the ability to deliver expert system

technology throughout the public and private sectors for a wide range of applications and diverse computing environments. CLIPS is being used by over 3,000 users throughout the public and private community, including all NASA sites and branches of the military, numerous federal bureaus, government contractors, 160 universities, and many companies.

2) CLIPS History

The origins of the "C" Language Integrated Production System (CLIPS) date back to 1984 at NASA's Johnson Space Center. At that time, the Artificial Intelligence Section (now the Software Technology Branch) had developed over a dozen prototype expert systems applications using state-of-the-art, hardware extensive demonstrations of the potential of expert systems. Few of these were put into regular use. This failure to provide expert systems technology within NASA's operational computing constraints could largely be traced to the use of LISP - the base language for nearly all expert systems software tools at that time. In particular, three problems hindered the use of LISP based expert systems tools within NASA: 1) the low availability of LISP on a wide variety of conventional computers; 2) the high cost of state-of-the-art LISP tools and hardware; and, 3) the poor integration of LISP with other languages (making embedded applications difficult).

The Artificial intelligence Section felt that the use of a conventional language would eliminate most of these problems, and initially looked to the expert systems tool vendors to provide an experts system tool written in a conventional language. Although a number of tool vendors started converting their tools to run in C the cost of each tool was still very high, most were restricted to a small variety of computers, and the projected availability times were discouraging. To meet all of its needs in a timely and cost effective manner, it became evident that the Artificial Intelligence Section would have to develop its own C based expert systems tool.

The prototype version of CLIPS was developed in the spring of 1985 in a little over two months. Particular attention was given to making the tool compatible with expert systems under development at that time by the Artificial intelligence Section. Thus, the system of CLIPS was made to closely resemble the syntax of a subset of the ART expert systems tool developed by Inference Corporation. Although originally modelled from ART, CLIPS was developed entirely without assistance from inference or access to the ART source code.

The original intent of the prototype was to gain useful insight and knowledge about the construction of expert systems tools and to lay the groundwork for the

construction of a fully usable tool. The CLIPS prototype had numerous shortcomings. However, it demonstrated the feasibility of the project concept. After additional development, it became apparent that sufficient enhancements to the prototype would produce a low cost expert systems tool that would be ideal for the purposes of training. Another year of development and internal use went into CLIPS improving it portability, performance, and functionality. A reference manual and user's guide were written during this time. The first release of CLIPS to groups outside of NASA, version 3.0, occurred in the summer of 1986.

Further enhancements transformed CLIPS from a training tool into a tool useful for the development and delivery of expert systems as well. Versions 4.0 and 4.1 of CLIPS, released in the summer and fall of 1987 respectively, featured greatly improved performance, external language integration, and delivery capabilities. Version 4.2 of CLIPS, released in the summer of 1988, was a complete rewrite of CLIPS for code modularity. Also included with this release was an architecture manual providing a detailed description of the CLIPS software architecture and a utility program for aiding in the verification and validation of rule-based programs. Version 4.3 of CLIPS, released in the summer of 1989, added still more functionality.

3)   CLIPS Version 5.0

Originally, the primary representation methodology in CLIPS was a forward chaining rule language based on the Rete algorithm (hence the Production System part of the CLIPS acronym). Version 5.0 of CLIPS introduces two new programming paradigms: procedural programming (as found in languages such as C and Ada) and object-oriented programming (as found in languages such as the Common Lisp Object System and Smalltalk). The object-oriented programming language provided within CLIPS is called the CLIPS Object-Oriented Language (COOL).

5-2.2   Knowledge Acquisition and Transformation

Knowledge acquisition refers to the transfer of expertise from human experts into a knowledge-base that will be used by an expert system to solve complex problems and provide decision making advice. It is commonly known that the manual knowledge acquisition process, which is widely used in the KBS development, involves the knowledge engineer conducting lengthy interviews with domain experts. Due to the complexity of various types of knowledge, the process of knowledge acquisition and knowledge transformation to the software coding is complicated and difficult. Sometimes

the results from the first round knowledge acquisition process are not satisfactory so it has to be repeated again and again until a satisfactory result is obtained. This "bottle neck" makes the entire process of KBS development time consuming and costly. To overcome difficulties associated with knowledge acquisition, it is important that the knowledge acquisition process be automated.

Promising and feasible solutions for easing the knowledge acquisition "bottle neck" do exist. Automated knowledge acquisition tools have been under research and development at Marquette University, and several other academic and industrial institutions such as Boeing Company. These automated tools will promote organized and structured knowledge elicitation, and are applicable to both declarative knowledge and procedural knowledge.

There are three automated knowledge acquisition tools that will be of potential use to develop a KBS system for the BP/VCWS. These are:

1)   A repertory-grid based automated knowledge acquisition tool, called the Expertise Transfer System (ETS) and its expanded version, AQUINAS. This tool was developed by Dr. John Boose at Boeing Advanced Technology Center, and has proven very powerful for increasing knowledge acquisition effectiveness. ETS and AQUINAS have been applied to various expert systems development projects, including several at NASA-Marshal Space Flight Center (MSFC);

2)   A Trait-Weighted Grid Knowledge Acquisition Tool for multiple-expert system development. This tool was proposed and developed by Dr. Xin Feng, the Principal Investigator of the ADCACS project at Marquette University. The main results extended the use of the repertory-grid method to the knowledge elicitation from multi-experts to build a multiple expert system for complicated situations (see Section 5-4.0 for reference).

3)   A computerized knowledge acquisition aid, called "Knowledge Acquisition Advisor" or "KA²". This IBM/PS2 based tool is capable of automatically generating and selecting appropriate questions that will be asked during the interview with domain experts. The answers from the domain experts will be directly input to the computer. Two significant features of KA² are:

   a)   A new knowledge concept, called the "knowledge cycle" is introduced and used to promote flexibility during the interview. The knowledge cycle is unlike other model forms because the design of the model is based entirely on the interpretation of the domain

problem by the Knowledge Engineer;

b)   The KA^2 has the potential of conducting "remote knowledge acquisition" (i.e., the face to face interview is not necessary). The interview may take place for domain experts who are far away from the development site. This is implemented by connecting two PCs by a Modem through a telephone line. The only cost for implementing this process is the telephone bill. This will reduce the cost of traveling and remove hassles for scheduling and other problems.

At the present time the KA^2 has developed to version 3.0, and has been tested at Marquette University by Dr. Xin Feng and several Research Assistants. The feedback from knowledge engineers about this tool is very positive.

5-3.0   Knowledge-Based System Control Application Items

The role of using a Knowledge-Based System (KBS), or an expert system for an intelligent control and decision support system has been specified in the above Section. The following functions, which are a subset of cupola functions and operations will be carried out by KBS:

1)   EVA and other operational monitoring in the Space Environment;
2)   Caution and warning systems (C&WS);
3)   BP/VCWS DMS System Testing and Diagnosis;
4)   Other emergency handling and decision support functions.

5-4.0   The Introductory Section of the Technical Paper:

Trait-Weight Grid Knowledge Acquisition and Inference Engine for Multi-Expert Systems in the Space Station Applications

Xin Feng
Department of Electrical and Computer Engineering
Marquette University
Milwaukee, WI 53233

5-4.1   Introduction

Expert systems encompass a wide variety of applications in different engineering fields and the current NASA space station. Generally speaking, an expert system consists of:

1)   a knowledge base containing problem solving knowledge elicited from the domain expert;

2)   a logic inference program called "inference engine" that performs inductive reasoning from the knowledge base to reach certain conclusions or decisions; and

3)   a user interface.

It is commonly known that knowledge acquisition, which refers to the transfer of expertise from human expert into the knowledge base, is the "bottle neck" for developing expert systems. This is due to the time consuming interview process between the knowledge engineer who is responsible for expert system development, and the domain expert who provides the knowledge. It is even more complicated to build an expert system based on multiple domain experts. Obviously, developing a multi-expert system requires eliciting knowledge from multi-domain experts and programming them into a comprehensive knowledge base.

There are advantages to developing and employing a multi-expert system. Most complicated problems in sophisticated environments, such as operations and controls, are expected to be solved not only by a single expert, but by a group of experts, usually called the consultation group. This group characteristically has a comprehensive knowledge background in multiple disciplines. Frequently, critical decisions are made by such multi-expert groups. It is also natural that domain experts often have different opinions or even conflicting viewpoints on certain aspects of the problem being considered, since each of them have different knowledge backgrounds and experiences,

and they view things from various angles. Further, an appropriate and effective inference engine is critical and essential for the multi-expert system to generate correct decisions or conclusions based on the comprehensive knowledge base. Obviously, more difficulties are anticipated for such a complicated task.

When conflicting opinions occur regarding a specific case in a decision making process, knowledge from each expert is not equally taken into account. Opinions from some experts are considered more heavily than others' opinions in the same consultation group. Final decisions may be made because they may have more experience, may be more familiar with the particular case, or simply due to their seniority. In other words, the reliability measure of knowledge from each expert should not be equal. Moreover, these measures are often vague and fuzzy, and new approaches, such as fuzzy logic, are more appropriate.

There are few efforts devoted to developing multi-expert systems. The principles of decision making processes in the inference engines were based on either "majority vote", or on primitive statistics. These methods, such as brainstorming, nominal group technique, and consensus decision making, did not consider the reliability measure of knowledge from each individual in the multiple expert group. This important measure is either neglected, or pre-biased without consideration of the real situation. Consequently, the resulting multi-expert system may not be appropriate. Wrong decisions and conclusions may be made. So far only little attention has been devoted to such a difficult, but essential issue in the expert system development.

Recently John Boose at Boeing Company proposed an effective knowledge acquisition approach, called "Repertory Grid" method, to perform knowledge elicitation and representation for a class of decisive type knowledge. One of the advantages of repertory grid technology is that it applies a scoring system to evaluate certain aspects of the specific knowledge by utilization of a two-dimensional grid composed of constructs (i.e. traits of problem solutions) and elements (problem solutions). Also, it provides a simple but effective approach to induce logical conclusions directly from the established grid without complicated logic programming and knowledge reconstruction. Several development tools, such ETS, PLANET (Gaines & Shaw, 1986), and AQUINAS (Boose, 1988) have been reported. Boose and his research group further considered the problem of expert knowledge measurement, and employed a mechanism to record the weights of each domain expert who provided the knowledge.

Despite the merits of this approach, the structure of the original two-dimensional grids do not reflect the degree of importance of the traits. Also, the reliability and quality measures of an expert's knowledge are not taken into consideration. In the case of a multi-expert environment, each expert only limits themselves to those traits that concern

him the most, without considering other traits when constructing such a repertory grid.

Motivated by the previous work, we propose in this report a new repertory grid method.This method employs a trait-weighted repertory grid for knowledge elicitation and inference engine development. This new construction strategy will enable each domain expert in the group to evaluate and justify their knowledge as represented by all traits instead of partial traits. Also, two new analytical techniques, the fuzzy entropy and set-valued statistics, are applied to accurately calculate the knowledge background and reliability measures of each individual domain expert in the multi-expert team. We also propose a new inference engine to derive the conclusions directly from the developed repertory grid representing the multi-expert's knowledge.

We discuss in this paper the repertory grid methodology and its principles. We then present the new structure of the trait-weighted repertory grid and the reliability measure of knowledge for domain experts in the multi-expert team. Later we will introduce the fundamentals of the trait-weighted grid, and derive two kinds of fuzzy entropy weights. Finally, we use a example of an space station emergency advisor multi-expert system to illustrate the effectiveness of the new approach.

CHAPTER SIX

STUDIES OF VIRTUAL DISPLAY GRAPHICS ENGINE ARCHITECTURE

6-1.0   Graphics Engine Architecture (GEA)

The basic architecture proposed for the graphics engine of BP/VCWS consists of two computational components. The LEEP format is used to represent Wide-Field-Of-View (WFOV) images.

The purpose of GEA is two-fold. First, the graphics engine displays stereo LEEP images taken from a closed circuit camera. Second, the graphics engine superimposes computer generated, "virtual reality" images onto the intended field-of-view in LEEP format. Three performance barriers have to be overcome for a successful GEA design:

1) Floating-Point Geometry Processing;
2) Integer Pixel Processing;
3) Frame-Buffer Memory Bandwidth.

The focus of this analysis is on floating-point geometry processing and pixel processing because of the intensive computations required for performing perspective and LEEP transformations. An excellent source on computer graphics can be found in [Fole90]. The research and investigations related to computer graphics and computer graphics design practices draw heavily on this source.

The representation of the proposed GEA is shown in Figure 6-1. The main processor specifies High-Level Graphical Primitives (HLGPs) as a part of its normal processing functions. The main processor communicates with the graphics engine through a shared memory. The Geometry Processor Component (GPC) of the GEA accesses the shared memory to obtain HLGPs for processing. The GPC processes the HLGPs by performing image perspective transformations and hidden edge and surface processing. Since stereo images are required, the GPC will process the image representation from two points of reference, presumably at points representing the eyes of the observer. As output, the GPC produces a stream of Low Level Graphics Primitives (LLGPs) such as lines, characters, and area primitives used by the Pixel Processing Component (PPC). A First-In First-Out (FIFO) memory or other type of memory may be necessary between the GPC and the PPC if the PPC is not sufficiently fast. As the output, the PPC produces a stream of pixels representing the image being rendered. The

stream of pixels are entered into the Frame Buffer Memory (FBM). An additional input to the FBM is an image produced by a LEEP video camera. The frame buffer is then used to drive a display.

Most computations expressed in this section are based on the assumption that the approach to rendering an image assumes a worst case analysis rather than average results.

6-2.0   A Study of Geometry Processing for the GEA

The amount of floating point arithmetic necessary to render an image depends on the approach chosen to display the image. The following analysis employs three dimensional images of triangles in space. Several assumptions are made regarding the processing of the image. First, no analysis is made regarding shading or other properties of the surfaces generated. Second, no clipping is performed in this analysis and assumes that the entire field-of-view is part of the view volume. Third, all geometries are recomputed for every image displayed. All computations described need to be performed 120 times per second.

The HLGPs can be lines, triangles, alphanumeric symbols, curves, and surfaces. The data structure for the line primitive includes two points representing the beginning and the end points of the line. The data structure for the triangle primitive includes the points representing the three vertices of the triangle as well as the equation of the plane in which the triangle resides. The plane equation helps the computation of triangle intersections. The data structure for the alphanumeric primitive includes a point representing the location of the symbol and a string representing a string of symbols. Additional information can be added to each primitive indicating the attributes of each primitive, as well as links between associated primitives where appropriate. The actual graphical primitives can be stored in memory in any convenient data structure, such as linked lists or arrays.

6-2.1   Study of Basic Image Transformation Algorithms.

The first part of the image rendering process performs a mathematical transformation on the image, so that the proper perspective of the virtual representation is constructed. The internal representation of the virtual reality can be composed in any convenient coordinate systems. One scenario is, that, if the user is walking through a static virtual reality, the possible changes to the view are rotations and translations due to the movements of the user and the distance between his/her eyes. Some kind of scaling will

also be necessary so that an image size will be compatible with the available display device. Thus, the virtual reality will have to be transformed and scaled to incorporate rotations with respect to the head and eyes of the user, so that the eyes will get the proper view.

These operations can be specified in terms of a matrix operation. Rotations, scales, translations, and perspective transformations can be performed in terms of matrix multiplications. In addition, the coordinate system is augmented with a fourth dimension. Two four-coordinate points are the same if they are multiples of one another, in all four coordinates. The new coordinate system is defined as **homogeneous** coordinate system as a result. Rotations, scales and translations do not affect this fourth dimension. Perspective transformations, however, will affect this fourth coordinate.

A translation in homogeneous coordinates can be represented by the following matrix:

$$T(d_x, d_y, d_z) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where d_x, d_y, and d_z are the desired translations in the x, y, and z directions. The translation of a point can be expressed as the matrix vector multiplication of the transformation with the homogeneous coordinate of the perspective point:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

Similarly, a matrix that can scale the coordinate system for a point can be represented by:

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where s_x, s_y, and s_z are the desired coordinate axis scalings.

Rotations may be implemented on each of the primary axis (i.e., x, y, and z axis). A matrix that rotates a point an angle θ_x about the x axis is given by:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Secondly, a matrix that rotates a point an angle θ_y about the y axis is given by:

$$R_y(\theta_y) = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Thirdly, the matrix that rotates a point an angle θ_z about the z axis is given by:

$$R_z(\theta_z) = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The perspective transformation assumes a normalized volume of,

$$\begin{aligned} -1 &\leq x \leq 1 \\ -1 &\leq y \leq 1 \\ -1 &\leq z \leq z_{min}, \end{aligned} \tag{7}$$

so that the volume associated with the virtual reality will be scaled with respect to all coordinate axes, as well. Perspective transformations can scale primitives relative to the distance from the view point. The perspective projection provides a means for properly determining hidden surfaces and primitives. The following matrix summarizes the perspective transformation for a three dimensional point:

$$
M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{1}{1+z_{min}} & \dfrac{-z_{min}}{1+z_{min}} \\ 0 & 0 & -1 & 0 \end{bmatrix}
\tag{8}
$$

where z_{min} represents the minimum distance from the point of perspective.

Each of the transformations can be represented as a matrix multiplication operation and can be preprocessed into a single matrix performing translation, rotation, scale and perspective so one matrix operation can do all necessary transformations, i.e.,

$$
P = T(d_x, d_y, d_z) S(s_x, s_y, s_z) R_x(\theta_x) R_y(\theta_y) R_z(\theta_z) M_{per}
\tag{9}
$$

where P is the transformation matrix for all points. The translation is, with respect to the center of the coordinate system, of an eye of the observer. The coordinate system scalings normalize the x, y, and z axis to the normalized view volume. The coordinate system rotations are performed based on the orientation of view with respect to the virtual representation of reality. The perspective transformation is performed with respect to the center of the coordinate system. P needs only be computed once for a particular scene since the matrix transformation will not change from point to point. Thus, the amount of arithmetic associated with transforming each point with P is 16 multiplications and 12 additions.

Assuming that 4,096 triangles are to be rendered, the amount of work necessary to perform this initial transformation, per image, is $4,096*3*28=344,064$ floating point operations. If 120 images per second are displayed, 41 MFLOPS (single precision) are required for this part of the rendering process.

6-2.2 Hidden Surface Determination

The next step in the process is the attempt to determine which primitives are visible, invisible, or intersecting. A partial ordering of the primitives will assist in this determination. Two triangles will not intersect if their ranges of x, y, and z do not overlap (the converse is not true, however). An example is shown in Figure 6-2, where the ranges of y overlap, but the ranges of x do not overlap so the triangles do not overlap. This observation can easily be extended to include the z dimension by noting that the triangle is in the x-y plane where z is constant. In this case, the ranges of y and z overlap, but the triangles do not. For those cases where triangles overlap in the x and y directions but not in the z directions, it is sufficient to draw the triangle that is furthest distance away first, and then draw the second triangle over the top of it in order to obtain the proper image rendering. The case where triangles overlap is considered later.

Determining the ranges of x , y and z for a triangle primitive requires 3 floating point comparisons per vertex or a total of 9 floating point operations. Thus, figuring out the ranges for all n triangle primitives would require 9n floating point operations. The comparison will be done according to the following partial ordering function:

$$\text{Compare } (x_L^1, x_H^1, y_L^1, y_H^1, z_L^1, z_H^1, x_L^2, x_H^2, y_L^2, y_H^2, z_L^2, z_H^2)$$

$$\text{if } \quad z_L^1 < z_H^2 \text{ then}$$
$$\text{return point 1}$$
$$\text{else if } z_L^2 < z_H^1 \text{ then}$$
$$\text{return point 2}$$
$$\text{else if } y_L^1 < y_H^2 \text{ then}$$
$$\text{return point 1}$$
$$\text{else if } y_L^2 < y_H^1 \text{ then }.$$
$$\text{return point 2}$$
$$\text{else if } x_L^1 < x_H^2 \text{ then}$$
$$\text{return point 1}$$
$$\text{else if } x_L^2 < x_H^1 \text{ then}$$
$$\text{return point 2}$$
$$\text{else}$$
$$\text{return tie.}$$

The comparison requires 6 floating point comparisons. If a tie exists, either the triangles do not overlap in a way that cannot be detected in the comparison, or the triangles overlap. An example of triangles that do not overlap, yet result in a tie, is shown in Figure 6-3.

Performing a partial ordering on the triangle primitives is a straight forward sorting algorithm requiring $O(n log_2 n) = 6n log_2 n$ floating point operations. Examples of such sorting algorithms can be found in [Bras88]. The estimated number of floating point operations required to process 4,096 triangle primitives is $9*4,096+6*4,096*12=331,776$ floating point operations. If 120 images per second are displayed, 40 MFLOPS are required for sorting the triangle primitives. Triangles that result in ties will be treated as a unit in the partial ordering, so that overlap computations can be more readily performed.

If two triangles intersect, it is necessary to compute the intersection and to transform the format into something that can be easily rendered. From the triangle primitives, the equation of the lines representing the intersection can be computed. The coefficients of the plane equations accompanying the triangle primitives, represent the normal for the plane. The vector representing the direction of the line of intersection is the cross product of the plane normals. The cross product computation requires 6 multiplications and 3 additions. Obviously, a vector direction *and* a point on the line are necessary in order to completely specify a line. Computing a point on the line requires 3 additions, 6 multiplications, and 2 divisions. If these floating point operations are normalized [Henn90], the set of computations requires a total of 26 floating point operations where computing time for one addition equals that of one multiplication and four additions equals one division.

The next step of the actual intersection of the triangles will be computed. This intersection can be imagined as that part of the line just computed that is common to both triangles. This operation can be performed by computing line intersections with each line segment of both triangles (six in total). It is not until this point that we can determine whether or not two triangles actually intersect. If the equation of the line is outside the bounds of both triangles, then the triangles do not intersect. This can be determined by comparing the intersection points with one triangle's vertices (9 floating point comparisons). In this case, if the triangles do not intersect, 115 normalized floating point operations are required. If an intersection is found, one triangle is broken into front and back parts. The triangle chosen is based on comparisons of the triangle vertices. A determination of which side one triangle vertex resides, with respect to the other triangle, requires evaluating the plane equations, a total of 3 multiplications and 3 additions per point. Thus, 36 floating point operations are necessary. The chosen triangle is then divided into three triangles for display. For intersecting triangles, a total of 151 normalized floating point operations are required. Assuming that 5% of the primitives do not intersect but cannot be detected by the first intersection test, and 5% intersect, the number of floating point operations required per image for intersection computations is 50,000 floating point operations. For 120 images per section, 6 MFLOPS are required.

6-2.3   GPC Risk Analysis

The computations proposed for the GPC requires a processing rate of 87 MFLOPS if an image composed of 4,096 triangles is rendered.

6-2.3.1   Risk of RISC Processor Computing Power

Implementation of the geometric transformations will require programming the processor to operate on the data structure corresponding to the image, and performance of the desired transformations. The analysis presented here indicates that, for the situation described, the computations are feasible and realistic.

Benchmarks based on the Intel i860 microprocessor indicate that the peak floating point rate was 80 MFLOPS in 1989, with a Linpack rate of 13 MFLOPS (double precision) [Fole90]. If Joy's Law [Katz90] holds true, by 1992, the required processing will be available. Joy's Law states that for RISC processors,

$$MIPS = 2^{Year-1984} \tag{10}$$

where 1 MIPS is an acronym representing one million of instructions executed per second. This law basically says the number of instructions that can be executed per second doubles with every passing year. Although MFLOPS and MIPS do not directly compare, a significant increase in performance may be possible, with a doubling in performance representing an upper bound. In addition, the MFLOPS rating of the Intel i860 is double precision. Single precision processing will, most likely, be faster. A risk in using the MFLOPS rating of 13 MFLOPS is that it only represents the performance of the Intel i860 when it is doing the Linpack benchmark and may not truly represent the computations discussed in this section.

6-2.3.2   Conclusion

From the discussions in the previous section, the use of a RISC processor comparable to the speed of Intel i860 will provide the computing power for the GEA.

6-3.0   LEEP Scan Conversion

The output from the GPC can be used as the input to the LEEP Scan Conversion Engine (LSCE). The perspective transformation performed on images applies the

appropriate perspective to all points, with respect to the z axis. Projecting the points onto a plane parallel to the x-y plane makes the scan conversion a function of the transformed points onto this plane and the LEEP transformation itself. Computations carried out in this section will map an image to a two dimensional pixel-based display device. Two display devices are required for stereo vision, however, the computations are identical for both cases, with the difference being the transformed primitives from the GPC. If the primitives are drawn in the reverse sorted order specified in §2,appropriate hidden edge and surfaces will be taken into account.

The LEEP transformation can be viewed as a mapping of the points on the projection plane onto the surface of a sphere of radius F and center at (0,0,0). The points on the sphere are then projected onto the plane z=0.

The raster scan conversion of line segments can be derived from looking at the equation of the line in space as well as the LEEP compression and resulting projection onto the view plane. Since the image was projected onto a plane with constant z, it is assumed that z=1 is a reasonable place for the projection plane. From the vertices of each triangle primitive, the equation of the line can be found. *All points considered in this discussion are the points in the transformed coordinate system.* The points will be considered in a spherical coordinate system, so that the symmetry of the spherical projection can be achieved.

6-3.1 LEEP Geometry

A line segment in the transformed coordinate system can be represented by giving two different points on the line starting at $(x_1,y_1,z=1)$ and ending at $(x_2,y_2,z=2)$. Assuming that the center of the field-of-view is at (0,0,0) and that the x and y planes correspond left/right and up/down respectively, gives an intuitive feel for the field-of-view being represented. The z plane corresponds to the field-of-view. The equation of the line can be expressed as:

$$y = mx + b, z = 1 \tag{11}$$

From the geometry of the LEEP compression and spherical coordinates,the spherical coordinates for a point can be computed. As a reference, θ is measured with respect to the z axis, and ϕ is measured with respect to the x axis on the x-y plane. Thus θ can be computed by:

$$\tan(\theta) = \sqrt{x^2 + y^2} \tag{12}$$

and

$$\theta = \tan^{-1}(\sqrt{x^2 + Y^2}). \tag{13}$$

Note that since $z=1$, the equation for the tangent function can be written solely in terms of x and y. Next ϕ can be computed:

$$\tan(\theta) = \frac{y}{x} \tag{14}$$

and

$$\phi = \tan^{-1}(\frac{y}{x}) \tag{15}$$

Using trigonometric identities, the values for other trigonometric functions of θ and ϕ are:

$$
\begin{aligned}
\sin(\theta) &= \sqrt{\frac{x^2 + y^2}{x^2 + y^2 + 1}} \\
\cos(\theta) &= \sqrt{\frac{1}{x^2 + y^2 + 1}} \\
\sin(\phi) &= \sqrt{\frac{y^2}{x^2 + y^2}} \\
\cos(\phi) &= \sqrt{\frac{x^2}{x^2 + y^2}}
\end{aligned}
\tag{16}
$$

The third component of the spherical coordinate system is the radius, R . The equation for R is:

$$R = \sqrt{x^2 + y^2 + 1} \tag{17}$$

The projection of this point onto the LEEP sphere is simply a coordinate with the same θ and ϕ. The radius of the sphere, F, serves as the third coordinate. The point $(x,y,1)$ is projected onto the sphere at the point (F,θ,ϕ). To complete the LEEP compression, the point (F,θ,ϕ) has to be projected onto the x-y plane. This projection can be done using trigonometry. First, the transformed coordinate system will be represented in coordinates (v,w) to differentiate from other coordinate systems. Roughly speaking, v will correspond to x and w will correspond to y. In terms of the point projected onto the LEEP sphere, the equation for v is:

$$v = F\sin(\theta)\cos(\theta)$$
$$= F\frac{x}{\sqrt{x^2+y^2+1}}$$
$$= F\frac{x}{\sqrt{x^2+(mx+b)^2+1}}.$$

(18)

Similarly, the equation for w can be found:

$$w = F\sin(\theta)\cos(\theta)$$
$$= F\frac{y}{\sqrt{x^2+y^2+1}}$$
$$= F\frac{mx+b}{\sqrt{x^2+(mx+b)^2+1}}.$$

Now both v and w are functions of x in the projected coordinate system.

6-4.0   Line Drawing in the LEEP Domain

Raster graphics is a process of mapping geometric objects onto a two dimensional display device. The display device is divided up into discrete picture elements (pixels). By turning the appropriate pixels on and off, an image can be represented on the display device. One problem with raster graphics approaches is determining which pixels to turn on for a given geometric object.

A line can be represented as a sequence of pixels on a display that are of a different intensity or color from its background. A scan conversion routine takes an arbitrary precision internal representation of a line, specified by its starting and ending points, and indicates which pixels should be affected by this line. The affect could be as simple as turning a pixel on or off, or specifying the color of a pixel. Due to the transformation specified in the previous paragraph, a straight line in the (x,y,z) domain will not be a

straight line in the (v,w) domain. To reduce the computational load, it would be more reasonable to compute pixels to be acted upon in the (v,w) domain because this domain limits the number of pixels. A full 3 dimensional image may require 100 times or more pixels to represent, depending on the physical distance from the origin.

Now that v and w are functions of x , the selection of pixels to turn on may begin. First, an initial point is selected. Usually this initial point is one of the end points of the line. The nearest point in (v,w) is lit. The next point to turn on is an iterative process of looking at perspective changes in v and w and seeing which is the best representation of the line. Therefore, given a starting point (v,w) , the following candidate points are to see which is the best one:

$$(v', w') = (v \quad , w + 1)$$
$$(v^2, w^2) = (v + 1, w + 1)$$
$$(v^3, w^3) = (v + 1, w)$$
$$(v^4, w^4) = (v + 1, w - 1)$$
$$(v^5, w^5) = (v \quad , w - 1)$$

$$(20)$$

These candidate points make the assumption that the line is drawn for increasing values of v. If the line has a different orientation, v and w may be swapped to reflect the appropriate orientation, so long as the overall coordinate system remains consistent (i. e. x and y would need to be swapped in response). Equation (20) also implies that values for (x_c, y_c) will be computed. The proximity of the candidate point (x_c, y_c) to an actual point on the line being drawn (x,y) will be the criterion used to select the pixel to be operated upon. The process of drawing points in the v-w plane continues until the end of the segment is reached. This can be determined by evaluating the corresponding ending point in the (v,w) plane. The algorithm can be summarized as follows:

for each line
> **compute starting and ending points (v_s, w_s) and (v_e, w_e);**
> **set (v,w) to (v_s, w_s);**
> **repeat until (v,w) equals (v_e, w_e);**
> **compute candidate points (v',w') from Equation (20);**
> **compute candidate points in (x',y') from Equation (11);**
> **select the closest point i to the line;**
> **set (v,w) to (v',w').**

Now that the basic algorithm has been presented, it is appropriate to tabulate the number of floating point operations that are necessary to complete an image. The tabulation will consider the amount of work necessary to draw one line, including some initial computations, in terms of the number of floating point operations required. The first

component is the initial work necessary to compute the starting and ending points of the current line in the (v,w) coordinate system. Computing the value of (vs, ws) from (xs, ys) requires one addition, four multiplications, one division, and one square root to compute (see Equations (18) and (19)). If these operations are normalized [Henn90], finding the starting point would require 14 floating point operations. For both starting and ending points, 26 floating point operations are necessary. For each point drawn, five candidate points need to be computed and evaluated. Computing x from a change in v requires putting Equation (18)in a different form, so that x can be solved. By rearranging Equation (18), a quadratic equation in x can be written as:

$$(v'+v'm^2-F^2)x^2+2mb(v')^2x+v'(b+1)=0 \tag{21}$$

where v' is a candidate for v. Computing the coefficients for this quadratic equation, and solving for x requires the equivalent of 26 floating point operations. Computing y from a change in w requires putting Equation (19)in a different form, so that x and then y can be solved. By rearranging Equation (19), a quadratic equation in x results:

$$(w'+w'm^2-F^2m^2)x^2+2mb((w')^2-F^2)x+(w')^2(b+1)-b^2=0 \tag{22}$$

where w' is a candidate for w. Computing the coefficients for this quadratic equation, and solving for x requires the equivalent of 27 floating point operations. Solving for y requires two additional floating point operations for a total of 29. Computing all candidates for x and y require 2*26+3*29=108 floating point operations. Finding the best match between all point and the line requires 5*(3)=15 floating point operations. In total, 123 floating point operations are required per point. In addition several floating point operations may be added to the required initialization, which increases the initialization computations to a total of 33 floating point operations. Therefore, the number of floating point operations required to draw a line in the (v,w) coordinate system is 123n+33 where n is the number of points ultimately drawn.

Assuming that a view is to be drawn as in Section 6-2. The view will consist of 4,096 triangle primitives. Therefore, 12,288 line segments are to be drawn. If each line will have five pixels, the number of computations required totals 8 million floating point operations. If two images are to be displayed, each at a rate of 60 times per second, the number of computations necessary is 960 million floating point operations per second.

Clearly, 960 MFLOPS is an unreasonable level of performance to expect. The algorithm specified above should be investigated more fully in an attempt to reduce the number of computations necessary per point. Indeed, if the computations necessary to figure out which pixel to light given a point (x,y) are accurate, a recursive line drawing routine may reduce the number of computations by a factor of two or more. The algorithm would work by breaking the line in half, and plotting the pixel corresponding to the mid-point. Each half is then recursively processed in the same fashion until already plotted pixels are encountered. It may be possible to compute the next point based on perhaps three rather than five candidate points as suggested by Equation ?.

In addition, there may be approximation approaches that result in good, yet unnoticeable, raster approaches. [Fole90] gives an excellent introduction to scan converting cubic splines. Some research will be necessary in order to find the appropriate splines that fit the LEEP distortion of lines.

Several image issues also need to be investigated. First, lines plotted as described in this section will appear ragged, and the raggedness may be very different between the two stereo images viewed. Investigation into anti-aliasing techniques may be necessary if a reduction in raggedness is desired. Another area deals with filling an area given the perimeter of that area. Virtual solid objects may have surfaces filled with an appropriate color or shade to impart more realism to the virtual reality.

6-4.1   PPC Risks.

The primary risk in the design of the PPC is being able to maintain the sustained rate of computations necessary to perform the LEEP transformation. From the computations shown, no single processor may be able to meet the computational requirements as described. Two directions for further action may be possible. The first is to research the LEEP transformations and attempt to reduce the computations either by approximation or by more clever algorithms. The second is to consider using multi-processing or custom VLSI.

6-4.1.1 RISC Processor

Due to the enormous number of computations required, it is unlikely that a single processor would be able to even come close to providing sufficient computational power. Even if the technology progresses optimistically according to Joy's Law, five of these processors would be necessary in order to meet the computational requirements, provided little processing power is lost due to having several processors in parallel. In

the worst case, perhaps ten to sixteen processors would have to be utilized, which is probably impractical due to the board space taken up. If more simplified raster approaches are discovered, this number could be reduced to a more reasonable level.

6-4.1.2 Custom VLSI

_   The VLSI system implements the suggested raster algorithm directly in hardware and exploits parallelism in some of the processing.

The basic building blocks for implementation are hardware adders, multipliers, dividers, and possibly square units. For LEEP rasterization, the most computationally intensive part of the process are the computations necessary to compute candidate points. The computations necessary for each point are identical, so it may be possible to assign processing circuitry to the computation of each candidate point, in order that work may be performed in parallel. Each computational unit would require each of the floating point hardware units previously described. The discussions that follow assume that computations are performed using 32 bit IEEE floating point numbers.

First a computation of the required hardware is performed. A carry look ahead adder can be constructed with approximately 2,000 transistors (assuming a CMOS implementation). (See [Poll90] for a treatment of carry look ahead adders and multiplication hardware.) The adder will have a gate delay of approximately 9 gate delays. Using a carry save multiplication technique, a multiplier can be constructed from approximately 25,000 transistors that will have a gate delay of approximately 45 gate delays. This gate delay assumes that partial products are added using carry look ahead adders. Assume further that division and square root units can be constructed using 40,000 transistors, and have propagation delays of 180 gate delays. If the prospective points can be computed in parallel, and if one nanosecond gate delay is assumed, the following expression results for the amount of time it takes to draw a line:

$$\text{Time} = 3*9\text{ns} + 14*45\text{ns} + 2*180\text{ns} + (8*9\text{ns} + 7*45\text{ns} + 3*180\text{ns})n$$
$$= (1017 + 927n)\text{ns} \tag{23}$$

It would take approximately 70 ms in order to render a single image consisting of 4,096 triangle primitives, where in order to render an image at 120 images per second only 8 ms time is allowed. However, the approach suggested above required 500 thousand transistors to implement (in CMOS). If as many as 1. 5 million transistors are available, then some performance improvement is possible. If three separate lines are

calculated simultaneously, an image can be rendered in about 24 ms. Further gains may be realized from pipelining. Whether or not sufficient improvement can be made is an area open to further investigation. In addition, other multiplication, division, and square root hardware implementations may be both faster and more efficient in utilization of transistors. Lastly, the three graphics processors may be operated in parallel in order to obtain the desired performance.

_ The risks associated with performing a VLSI implementation of the rasterization are high; because of the level of integration suggested and the performance computations.

With the progress achieved in VLSI systems, and their history, the technology should be available in the 1993 time frame. Some technology problems could occur due to problems making the graphics VLSI systems work in parallel.

The cost risks are that the suggested VLSI system may be very expensive to develop and manufacture because of its size. The development costs may be added by the regular structure suggested in this section. Because of the level of integration, yields may be small. In addition, the production runs may be small, further driving up the cost associated with the systems.

The biggest risks in the schedule would be in the design and specification of the VLSI system. This process may take one man-year or more.

6-5.0   Frame-Buffer Memory Requirements

Some simple computations considering memory bandwidth can be performed in regards to the FBM requirements. If each image is $10^{\wedge}6$ pixels, three bytes are allocated per pixel (a byte for each primary color), and sixty images are displayed every second. The memory bandwidth required per image is approximately 377 million bytes per second, assuming the display device will read all pixels. Two images required for stereo viewing would increase this data rate to 754 million bytes per second. This requires an average access time of 1.3 nanoseconds per byte.

Clearly, some effort has to be expended in the appropriate design of the memory system. Either a wide or interleaved memory architecture would be necessary in order to improve the bandwidth. The frame buffer memory would require at least 6 megabytes of memory. Assuming an interleave of 8, eight 1 meg by 8 banks of such memories would be necessary and each memory bank would require a sustained 10 nanosecond access time. The memory banks can be interleaved further with eight 256k by 8 memories, each with an access time of 80 nanoseconds. If DRAMS are used, 256k by

4 memories with 80 nanosecond access times are readily available in the current technology. Thus, the frame buffer memory would require 128k-256k by 4 memories in today's technology. A further advantage of the 256k by 4 memories is that sufficient memory is available for double buffering.

In 1993 technology, 128k by 8 SRAMs may be available [Henn90], which could result in faster access times and simpler memory design. If double frame buffering is desired, the required memory bandwidth can be halved by writing into one buffer and displaying from the other. This can be accomplished as above using 128k by 8 SRAM memories.

Further investigation and technology advances may result in more reasonable number of devices necessary to implement the FBM. Also, the analysis in this section only considered average access time and assumed that the video imaging interacted without interfering the PPC. Therefore, peak traffic and interference may place greater demands on the FBM. Ideally, the imaging device would place its image into the FBM, then the PPC would superimpose the virtual reality on top of the image. The actual situation may be very different. In addition, the memory bandwidth can be reduced by a factor of three by only considering gray scale images. An added advantage is that the memory capacity can be reduced by this factor, as well. Lastly, it may be possible to set up an additional frame-buffer, so that the imaging device writes into one, the PPC draws lines into the next, and the display device reads from the third. This triple buffering simplifies the FBM interfaces at the expense of additional memory devices.

6-5.1 FBM Risks

Two risks are possible in the design of the FBM. The first relates to performance. Can the FBM provided sufficient bandwidth to support the required memory access patterns? In terms of the design, a FBM can be designed with the desired characteristics. The second, and more difficult risk,will be keeping the amount of circuit board space to a minimum, so that sufficient memory can be made available.

6-6.0 Overall Assessment of Risks

In this section, an overall assessment of the risks associated with the GEA analysis is discussed. Other discussions can be found with discussions of each of the three components of the GEA. Several different architectures are considered and evaluated. The architectures can be interpreted as preliminary investigations into the implementations.

6-6.1 Parallel Computing Architectures

Parallel computing architectures can be used to speed up processing by employing several computers working together simultaneously. Parallel computing approaches are a possible approach for managing the computing necessary to raster the image. Multiple instruction multiple data (MIMD) [Henn90] has several computers running their own program on their own data. A systolic architecture is a computer system set up as a single instruction multiple data SIMD computer system. Several processors are interconnected in a regular mesh, so that information and partial results can be communicated. All processors are running the same instruction sequence. The systolic processors can either be very simple (bit serial processors) or as complicated as a typical microprocessor [Foul87, Fort87].

6-6.1.1 Possible Implementations

Because of physical design limitations, no implementations are suggested.

6-6.1.2 Analysis of Risks

In general, the overall risks of using parallel processing to manage the complexity of the image rastering process are high. This can be stated as follows:

1) Technology Risk

The technology risk is very high. The current state of the art has parallel computing architectures that can be implemented on the board level. An excellent summary of parallel processing technology can be found in [Dunc90]. If advances in the scale of circuit integration [Henn90] hold true,in the 1992-1993 time frame, perhaps 4 to 8 processors may be integrated on the same chip, not withstanding chip bandwidth issues. At the current time, systolic architectures have been implemented that have sustained processing rates of the order of 1 billion floating point operations per second [Foul87]. Because of physical design constraints of the body ported workstation, this design is impractical. As far as more general parallel processing machines, the Monarch parallel processor [Rett90] is capable of processing 2 billion floating point operations per second. This amount of processing would require eight processor boards, four memory boards, and one utility board. This clearly is even more outside the realm of practicality. If this implementation direction were chosen, the technology would need to develop to the point where the circuitry integrated using VLSI today, is integrated with at least ten times the current circuit densities.

2) Cost Risk

The cost risk for development is moderate, if based on current technology. If new technologies are desired, the risk increases significantly.

3) Schedule Risk

The schedule risk for the developing and delivering this technology is very high. The high risk comes from a dependence on the technology. If the technology is not available, then the development schedule will suffer.

6-6.2   RISC Architectures

A RISC processor can be used in each of the two parts of the GEA. For the image geometry computations, a RISC processor has a moderate but acceptable risk. For rastering, the RISC processor is probably not an acceptable risk.

6-6.2.1   Current State of the Art

The current state of the art RISC processor is the Intel i860. It has a peak processing rate of 80 MFLOPS and a sustained rate of 13 MFLOPS. If Joy's Law holds, 8 to 16 times this power will be available by 1993. The Intel i860 also has certain computer graphics hardware built onto the chip. Memory architectures have to be developed to allow the processor to sustain this rate of computation. Current memory technology is available that can fulfill the need, however,such memory architectures are expensive in terms of the board space they require.

6-6.2.2   Analysis of Risks

The risks associated with utilizing RISC processors are moderate. For the front end of the GEA, the risk appears to be very manageable. For the back end, the risk appears to be moderately high. If more efficient rastering techniques are found, the processing would be manageable. Maintaining appropriate memory bandwidth while maintaining a small size may be a high risk, due to the necessity of interleaving memories.

1) Technology Risks

The technology risk is moderate for the GPC and very high for the PPC. The primary technological risk is making the GEA work at an acceptable level of performance. Two risks are possible. First is the technology associated with rastering the compressed image efficiently. Second is making the two stage GEA work together efficiently. If a parallel processing approach is taken for the back end of the GEA, a possible risk is not being able to make the processors work together. Computations performed for the PCC indicate that the processing power from such a RISC processor may not be sufficient.

2) Cost Risks

The cost risks are low. The RISC processors are currently available, as are the memory devices. One cost risk that is difficult to quantify is the cost associated with reducing the number computations necessary to raster the image. However, this risk is balanced by the potentially large payoff of a more reasonable rastering technique.

3) Schedule Risks

The schedule risks are moderate. Schedule risks include problems encountered in reducing the raster computations or difficulty encountered in making the parallel RISC processor work. The processor discussed in the discussion of the GEA is currently available. This approach has a RISC processor front end that will perform geometric transformation and a back end that will be implemented as a VLSI system.

6-6.3 Custom VLSI

A custom VLSI system has been proposed as an approach for implementing the LEEP raster graphics.

6-6.3.1 Current State-of-the-Art

At the present time, about one million transistors may be placed on a single VLSI system, and, perhaps as many as 1.5 million may be possible by 1993 [Henn90]. The VLSI system will most likely be a state of the art system in terms of the size of the integration.

6-6.3.2 Analysis of Risks

The risks associated with custom VLSI include the design time required, as well as being able to meet the required processing capability. The architecture will require on the order of 500,000 transistors. For the processing required, this architecture may be integrated three times on the same chip, totaling 1. 5 million transistors. This is near the state of the art projected for 1993.

1) Technology risks

The technology risks are moderate, given the past history of VLSI circuit densities.

2) Cost risks

The cost risks are high due the high cost of doing state of the art VLSI design and fabrication.

3) Schedule risks

The schedule risks are moderately high. This is because of the design time required for the VLSI device, which may require a man-year or more, for the design alone.

6-6.4 Frame-Buffer Memory

The risks associated with the FBM are twofold. The two risks are supplying the required memory bandwidth and minimizing the amount of board space.

6-6.4.1 Current State-of-the-Art

Current memory technology is capable of supplying either the capacity or the speed with the current technology. The problem is that high capacity, high speed memories are currently limited to 256k bit range. Devices in the 1M bit range may be available in the 1993 time range. The problem with these devices is that as many as 128 of these devices may be necessary to support the FBM and double buffering. Slower, 16M bit memories may be available in the 1993 time range, but may not be fast enough [Henn90].

1) Technology Risks

The technology risks are moderate to high. Either of the design goals can be met if the other is ignored. When combining the two design goals, the problem is much harder.

2) Cost Risks

The cost risks are moderate to high. The cost comes from having to either provide high capacity very high speed memory devices or to take existing memory devices and use special packaging technologies.

3) Schedule Risks

Schedule risks are high. The main schedule risks result from the possibility of having to research a new packaging technique, if high capacity, high speed memory devices are not available.

6-7.0   Conclusions and General Recommendations

The general recommendations for the GEA are to implement the geometric processor using a RISC processor such as the Intel i860, and pursue research investigating the scan conversion of the LEEP transformation. If the computations are reduced, perhaps an Intel i860 may be able to provide sufficient computational power. Otherwise, custom VLSI may be necessary in order to provide sufficient computational power. At least initially, both custom VLSI and LEEP scan conversion approaches should be investigated to see which direction is more practical. A couple of good starting points for finding better rasterization technique can be found in [Hobb90] and [Fole90].

Recommendations in regards to display technology are stated in Section 1 of this chapter Four.

It is also recommended that an emulator for the GEA be constructed with technology available today. Parallel processing and memory architecture design can be designed using the currently available memories and processors. Appropriate display devices also should be selected.

6-8.0   References

[Bras88]   Giles Brassard and Paul Bratley, *Algorithmics: Theory & Practice*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988. [Dunc90] Ralph Duncan, "A Survey of Parallel Computer Architectures," *IEEE Computer*, vol. 23, no. 2, February 1990, pp. 5-16.

[Fort87]   Jose A. B. Fortes and Benjamin W. Wah, "Systolic Arrays: A Survey of Seven Projects," *IEEE Computer*, vol. 20, no. 7, July 1987, pp. 91-103.

[Foul87]   David E. Foulser and Robert Schreiber, "The Saxpy Matrix-1: A General-Purpose Systolic Computer," *IEEE Computer*, vol. 20, no. 7, July 1987, pp. 35-43.

[Fole90]   James D. Foley, Andries van Dam, Steven K. Feiner,and John F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley,Reading, Massachusetts, 1990.

[Grim89]   Jack Grimes, Les Kohn, and Rajeev Bharadhwaj, "The Intel i860 64-Bit Processor: A General Purpose CPU with 3-D Graphics Capabilities, " *IEEE Computer Graphics and Applications*,vol. 9, no. 4, July 1989, pp. 85-94.

[Henn90]   David A. Patterson and John L. Hennessy, *Computer Architecture: A Quantitative Approach*,Morgan Kaufmann Publishers, Inc, San Mateo, CA, 1990.

[Hobb90]   John D. Hobby, "Rasterization of Nonparametric Curves," *ACM Transactions on Graphics*, vol. 9, no. 3, July 1990, pp. 262-277.

[John89]   Barry W. Johnson, *The Design and Analysis of Fault-Tolerant Digital Systems* ,Addison-Wesley, Reading, MA, 1989.

[Katz90]   Randy H. Katz and John L. Hennessy, "High Performance Microprocessor Architectures," *International Journal of High Speed Electronics*,vol. 1, no. 1, 1990, 1-17.

[Poll90]   L. Howard Pollard, *Computer Design and Architecture*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[Rett90]   Randall D. Rettberg, William R. Crowther, Philip P. Carvey,and Raymond S. Tomlinson, "The Monarch Parallel Processor Hardware Design," *IEEE Computer*, vol. 23, no. 4, April 1990, pp. 18-30
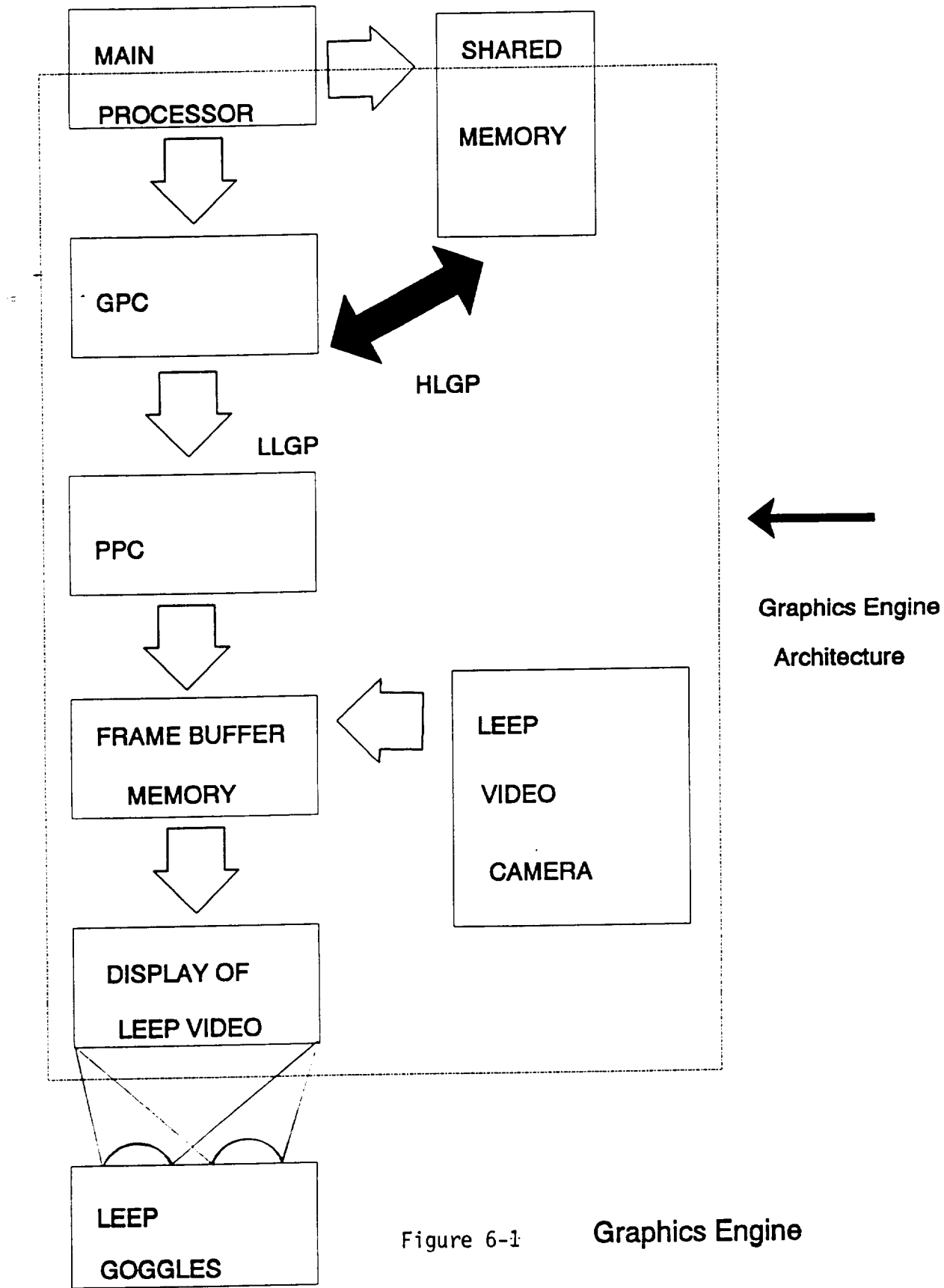
Figure 6-1   Graphics Engine
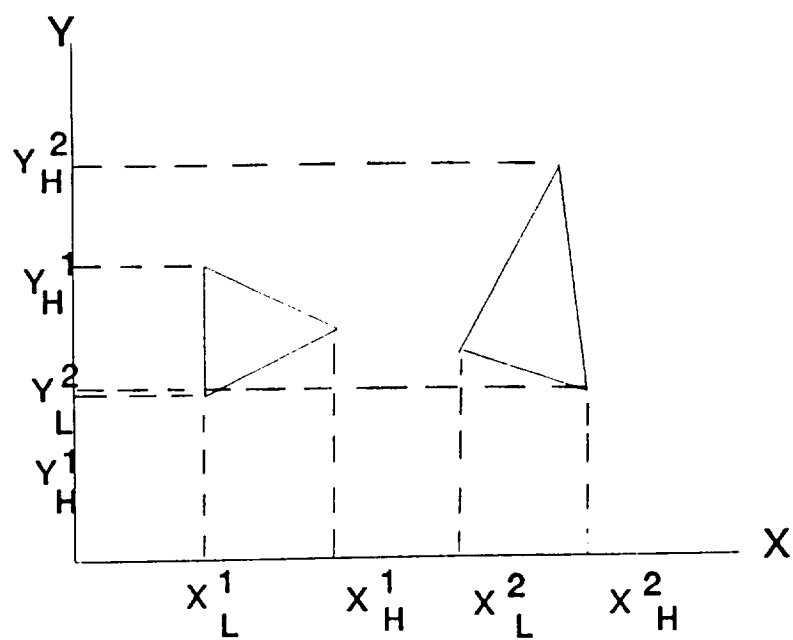
Figure 6-2   Simple Case of Non-intersecting Triangles.
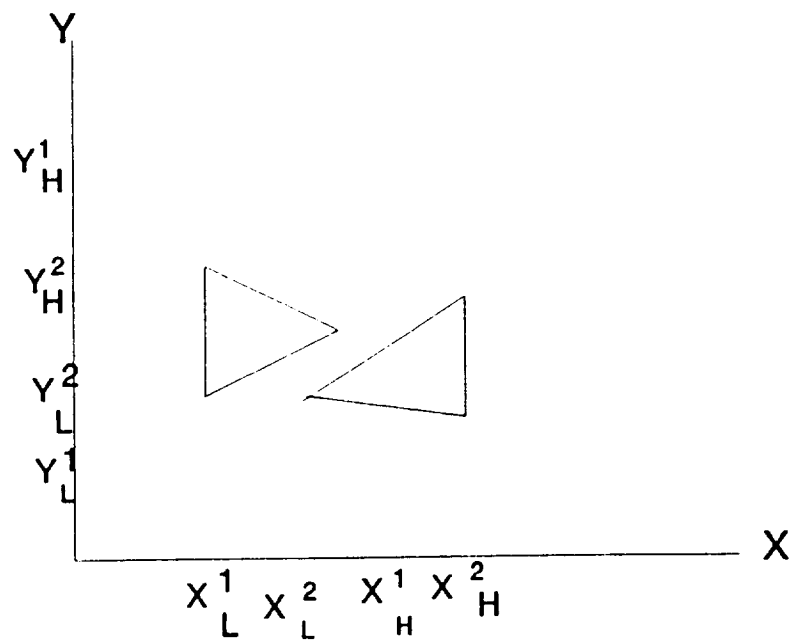
Figure 6-3   Difficult Case of Non-intersecting Triangles.
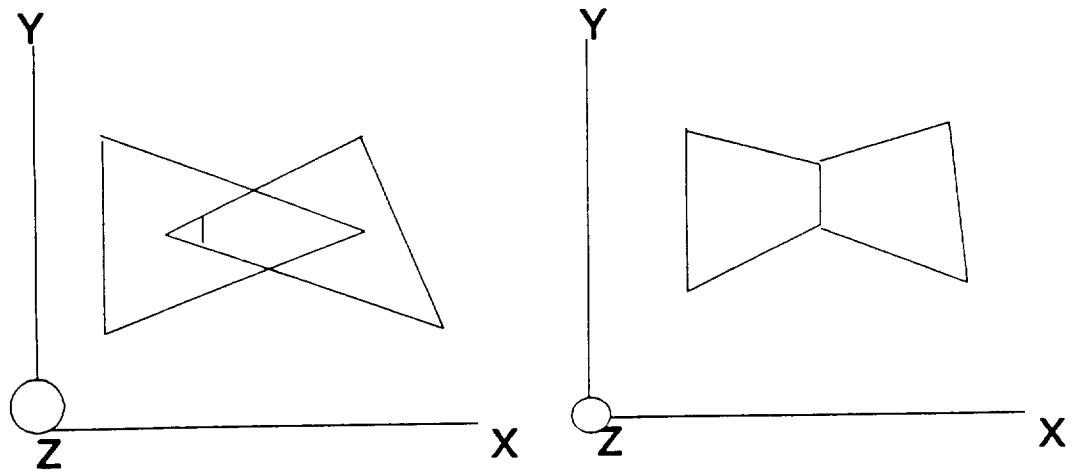
Figure 6-4   Triangle Intersection.

3D Representation of Object

Perspective Projection onto this plane is performed

View Plane

LEEP Sphere

P₂

P₁

Stereo LEEP Compressed Images

X-Y Plane

P_1= EYE 1

P_2= EYE 2
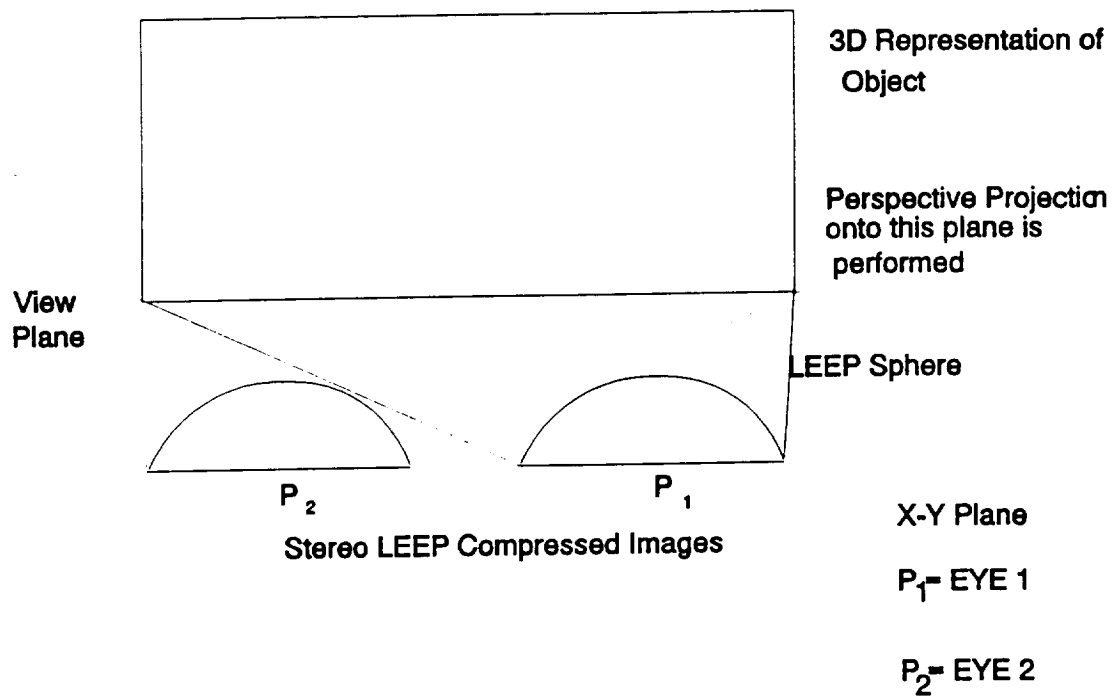
Figure 6-5   LEEP Transformations.

CHAPTER SEVEN

HUMAN FACTORS RESEARCH AT ACT LAB

7-1.0   Human Factors Research

Chapter Seven contains research and analysis of human factors that are carried out at the Marquette University Advanced Control Technology (ACT) laboratory. The information is believed to be useful in the design and analysis of the experimental setups.

It is suggested from the research that the eye-tracking technology is not an approprate approach to be employed in the BP/VCWS, due to the irregular movements of the eyes and other human factors.

7-2.0   Stereoscopic versus Monoscopic Displays

There has been research done on the need for stereoscopic displays to achieve optimum human performance in the workstation environment. Stereoscopic displays have been found to enhance form recognition as well as contrast sensitivity when it is added to the monocular cues. Stereoscopic displays improved cursor positioning in a 3D visual presentation. The most prominent advantage for stereoscopic display is in a finding by Yei-Yu Yeh at Honeywell Systems and Research Center Phoenix, AZ, which states that mean reaction times were significantly faster with the stereoscopic displays than with the monoscopic perspective presentations in most of his experimental conditions (5 of 6). Yeh also found that when objects were located close to the viewing reference point, monocular depth cues were acceptable for depth judgement. But when the object was viewed at distances further from the viewing reference point, the monocular depth cues were insufficient. When the same situation was viewed through stereoscopic perspective, the benefits of stereoscopic presentations become progressively larger.

7-3.0   Eye Movement and Eye Tracking

The eye tracking idea for the workstation to be designed has been investigated. The research on types of eye movement, speed, frequency and consciousness of movement of the eye was completed by the end of October. If eye movement can be better understood, attempts can be made to calculate the target that the eye will next focus on given initial velocity and angle. Research was also conducted on the different

types of eye tracking devices that are in use today.

The maximum velocity of the eye was found to be 21.43 deg/sec for a lateral fusion movement of 5.5deg. (i.e. a horizontal movement left or right in the same distance plane). This is much slower than saccades. Saccades are another type of eye movement that are determined to be involuntary, yet it reaches maximum velocities of 500 and 700 deg/sec, have an amplitude of 20 deg. Two or three occur every sec.

These movements in magnitude and velocity must be taken into account when designing an eye tracker to insure sufficient sampling frequency.

Retinal habituation, "the tendency of neural cells to become less sensitive when there is continual sensory input," is a phenomenon that must also be managed. Because of this phenomenon, the ocular motor system allows the eye to slowly drift so that the retina can be continually updated. These slow drifts have an amplitude of 5 min of an arc with a frequency of 5 Hz and an angular velocity of 1 min of arc/sec.

The information that was found on the eye movements showed a great diversity in the types of movement and their speed and frequency. There were also measurements taken for rapid flicks of the eye as well as other irregular movements that did not fit into the previously mentioned categories.

The eye tracking techniques that were investigated included the search coil method as well as the use of infrared light. The search coil techniques required that a small coil be attached to the eye -- there were a variety of different ways to do so -- but this requirement was the biggest problem for this project. The coils could not be connected to the subject's eye for an extended period of time without some discomfort. This technique involves applying a magnetic field about the eye, and by the fluctuations in the field's magnitude and orientation caused by the moving coil on the eye, the direction in which the eye was focusing could be determined. One problem of the search coil method is the fact that some acuity is lost due to the irregularity of the coil on the surface of the eye.

The next technique required focusing an infrared beam of light on the eye, and by measuring the amount of reflected light, the position of the eye could be determined. This procedure allows the subject the most comfort because nothing is in direct contact with the eye. The problem with this device is that as the current design of the work station is to include the LEEP optics optical system, the focal length is too small (38 mm) for the placement of infrared light sources, without even considering a device to measure the reflected light.

From this information on the eye tracking devices and the obvious difficulties involved in tracking the eye with its very irregular movements, it has been concluded that another means to orient the user in the virtual world must be found.

7-4.0   Field-of-View

The movement of the eye is found in the Military Standards 1472 as being 35 deg left and right and 40 deg up and 20 deg down (maximums) while holding the head still. These numbers do not appear to include peripheral vision. If included, these numbers would increase significantly. Sources have been found that give 180 deg horizontal and 150 deg vertical field-of-view.

The field-of-view maximums are important because they give an idea of the amount of space in which information can be displayed. Another aspect that should be considered is the fact that the density of the information displayed plays an important role in the useful field-of-view. "The useful field-of-view constricts when there is high density of detail to be processed by eye and brain." When there is a visual overload of information, the eye/brain reacts in a series of different ways: the useful field-of-view is narrowed to adjust for the amount of information being processed, the visual scan is slowed due to more and smaller movements of the eye and longer fixations, and accuracy will also be diminished due to the overload. These phenomena must be kept in mind when designing what the user will see on the display.

7-5.0   Resolution of the Eye

The resolution of the eye can be determined by the dimensions and location of the cones on the retina. The center-to-center distance of human cones is 2.02 to 2.32 um. The cones have a diameter of 1.5um.

At viewing distance (2 feet) for a 1280x1024 pixel screen, each pixel is about 2 minutes of the visual field. This will greatly change when the display is moved to with inches of the eyes. The human eye can distinguish detail down to 1 min of an arc.

Once an appropriate mapping function is found, this data can be used to determine the absolute maximum bandwidth necessary for human display presentation.

7-6.0   Blind Spot

The blind spot of the eye is where the optic nerve enters the retina and no cones or rods are located. The image that is projected on this area can not be perceived.

For this project, it was first thought that calculations would not have to be done to produce an image at this area because it would not be seen. This was considered and then dismissed because the calculations to determine the image's continual location would be more extensive that the calculations to form the picture in that area. This would also require eye tracking.

However, the one thing that should be kept in mind is that when location of warning signals and other important data is being considered, the locations should not coincide with the blind spot, if the subject is focusing on the main or center part of the display.

7-7.0   Acuity

The eye automatically functions to allow the greatest amount of light through to insure the best acuity. Larger pupils allow the greater angles of light to strike the cones. The relative acuity of vision in the central and peripheral fields of the retina has been found in the literature.

7-8.0   Perception

Choosing the view point and field-of-view are important options that must be considered when designing the displays to be shown to users of the body ported work station. Optimal choices for this project would be a choice of view point that is at the eye, and a field-of-view which is wide angle. Papers have been found that quote normal human binocular field-of-view to be 120 degrees horizontally and vertically with up to 90 degrees overlap in the binocular field. These design features would allow for the best case solution for the project. The problem then becomes displaying the three dimensional virtual world in two dimensions in such a way that the reconstruction to three dimensional understanding is perceived correctly. The reconstruction process is based largely on the location of view point, size of field-of-view as well as the level of detail of the objects. Assumptions by the user determine the values of the parameters, which results in distortion and subsequent errors in judgements.

Familiarity of the objects displayed, greatly aid the user in the interpretation of the scene. Consistent characteristics of an object like parallelism and perpendicularity of planes or lines are essential for the reconstruction of that object from a two dimensional perspective display.

To minimize the error of interpretation that could occur, the center of projection of the display should be placed slightly in front of the eyes to achieve a wider field-of-view than would normally be calculated by the normal geometry. A grid in the ground plane

of the virtual world will minimize errors in judgements of distance; another helpful aspect of three dimensional virtual displays would be to extend a line connecting objects that are suspended above the ground plane to the ground plane.

The above information will be implemented, but it is assumed that all figures can be graphically depicted in complete form. If complete objects can not be displayed because of the speed limitations of the graphics generator, then important choices must be made as to the parts of the object that should be displayed. The parts of a figure that contain the most information are the locations where a number of lines or planes converge and the convex and concavities of the outline of the object to be displayed. These features allow for the recognition of a specific object and the ability to distinguish it from other objects. The line or planes of an object that are parallel are not the most critical aspects of the display. Additional research is being done in this area to insure that the displays designed for this project will not be misinterpreted.

CHAPTER EIGHT

LITERATURE SEARCH AND PARADOX DATABASE LISTINGS

File ID: 00001

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:
Key Word 4:

Author: Westheimer, Gerald
Title: Oculomotor Control: The Vergence System

Media: Eye Movements and Psychological Processes

Publication Date: 9-Apr-76
Publisher, City: Lawrence Erlbaum Assoc., Hillsdale, NJ
Page #:   40, 41
Document ID: ISBN 0-470-15029-7
Agency: US Army
Type of Document: Symposium 4/15/74

Abstract/Citation:
  p.40
  2. A relatively long delay (200-250 msec) elapses between the
visual stimulus and the resulting saccade. About 150 msec can be
accounted for by the time it takes visual information to reach the
cortex (about 80 msec) and the latency to elicit a saccade by
cortical stimulation (about 70 msec)
  p. 41
The maximum velocity achieved during a human saccade increases
with amplitude up to 20 degrees; for larger saccades, a velocity
saturation occurs at between 500 and 700 degrees/sec

File ID: 00002

Subject: Human Factors
Key Word 2:   Eye Tracking
Key Word 3:
Key Word 4:

Author: Anliker, James
Title: Eye Movements: On-Line Measurement, Analysis, and
        Control
-Media: Eye Movements and Psychological Processes

Publication Date: 9-Apr-76
Publisher, City: Lawrence Erlbaum Assoc., Hillsdale, NJ
Page #: 187
Document ID: ISBN 0-470-15029-7
Agency: US Army
Type of Document: Symposium 4/15/74

Abstract/Citation:
    p.187
    Our plans for the development of PERSEUS call for the inclusion
of an advanced eye tracker as a source of eye-position signals. The
accurate two-dimensional eye tracker developed by Cornsweet and
Crane (1973) has been using a simpler, less accurate, eye-movement
monitor (Biometrics) as a sort of stand-in capable of generating
real-time electrical signals corresponding to vertical and
horizontal components of eye movements. We selected the
Cornsweet-Crane eye tracker for this major front end role in PERSEUS
besause it is a)sufficiently accurate (better than 5 min of arc), b)
insensitive to translational movements of the eye, c)convenient and
noncontacting (uses reflected images form the eye), d)noninterfering
(uses infrared light), e)capable of nearly linear response over a
sufficiently wide angle...

File ID: 00003

Subject: Human Factors
Key Word 2:   Field of View
Key Word 3:
Key Word 4:

Author: .Mackworth, Norman H.
Title: Stimulus Density Limits the Useful Field of View

~Media: Eye Movements and Psychological Processes

Publication Date: 9-Apr-76
Publisher, City: Lawrence Erlbaum Assoc., Hillsdale, NJ
Page #:   309, 312
Document ID: ISBN 0-470-15029-7
Agency: US Army
Type of Document: Symposium 4/15/74

Abstract/Citation:
    p.309
    ...useful field of view for these subjects was about 6 degrees on either side of the central fixation point. The limit was set by the small size of the targets and by their similarity to the nontarget circles.
    p.312
    ...the useful field of view is the area around the central fixation point that is being effectively processed in a single fixation. Two quite different procedures give the same estimate of the size of the useful field of view. The size varies with the density of the material which is being processed. With the dense displays, the useful field of view appears to about 1 degree across...

File ID: 00004

Subject: Human Factors
Key Word 2:   Eye Recognition
Key Word 3:
Key Word 4:

Author: Gould, John D.
Title: Looking at Pictures

Media: Eye Movements and Psychological Processes

Publication Date: 9-Apr-76
Publisher, City: Lawrence Erlbaum Assoc., Hillsdale, NJ
Page #: 324
Document ID: ISBN 0-470-15029-7
Agency: US Army
Type of Document: Symposium 4/15/74

Abstract/Citation:
   p.324
   Wiliams (1967) showed how the color, shape, and size of objects on a display affected whether or not the objects would be fixated. A key result was the color greatly influences subject's fixation patterns whereas size and shape do not. Color provides a good cue for subjects to perform grouping operations on figure-background relations, whereas shape of size of objects evidently are not as effective. More generally, color coding is an effective means for people to locate targets quickly.

File ID: 00005

Subject: Human Factors
Key Word 2:   Visual Cognition
Key Word 3:
Key Word 4:

Author: Humphreys, Glyn W.
Title: Visual Cognition: Computational, Experimental and
        Neuropsychological Perspectives
-Media: Book

Publication Date: 1-Jan-89
Publisher, City: Lawrence Erlbaum Assoc. Ltd., East Sussex,
Page #:
Document ID: ISBN 0-86377-124-6
Agency:
Type of Document: Book

Abstract/Citation:
    p.74
    [looking at the reversing faces-goblet figure] When the figure is
either of the faces, then the concavities pointing into them define
th eparts as forehead, nose, shin, etc. This demonstration can help
explain how the same contour can be "recognized" as two distinct
objects: what matters is the way in shich the contour is partitioned
prior to recognition, and this in turn seems to involve a simple
search for concavities referred to the centre of whichever region is
seen as figure.
    p.76
    Biederman argues that there are a number of "non-accidental"
properties of edges in images which can be used as reliable cues to
related properties of edges in the world. Such properties include
collinearity, curvilinearity, symmetry, parallelism, and
co-termination.
    p.77
    See Fig. 3.19
    [Caption] Objects are much more easier to recognise when their
contours are degraded in ways which preserve their concave part
intersecions than when contours are deleted at regions of concavity

File ID: 00006

Subject: Human Factors
Key Word 2:   Blind Spot
Key Word 3:   Eye Diagram
Key Word 4:

Author: Begbie, Hugh G.
Title: Seeing and the Eye: An Introduciton to Vision

~Media: Book

Publication Date: 1-Jan-69
Publisher, City: Natural History Press, Garden City, NY
Page #: 30,103
Document ID:
Agency:
Type of Document: Book

Abstract/Citation:
    p.30
    See Fig. 13. [Caption] Horizontal section of the eye.
    p.103
    See Fig. 25. [Caption]The relative acuity of vision in central
and peripheral fields of the retina. The acuity of the central
fovea has been taken as 1. The solid line represents acuity of cone
vision (light-adapted eye), and the dotted line represents acuity of
rod vision (approximate). The black area is the blind spot.

File ID: 00007

Subject: Human Factors
Key Word 2:   Eye tracking
Key Word 3:   Visual Cognition
Key Word 4:   Eye Resolution

Author: Foley, James D.
Title: Interfaces for Advanced Computing

Media: Scientific American

Publication Date: 1-Oct-87
Publisher, City:
Page #: 127-135
Document ID:
Agency:
Type of Document: Magazine

Abstract/Citation:
    p.128
    Artificial realities have three component: imagery, behavior, and
interaction....the user interacts with an artificial reality in much
the same way as he interacts with the three-dimensional world: by
moving, pointing, and picking things up, by talking and observing
from many different angles. [Caption]Graphcs Displays projected
inside the helmet shown on page 126 include menu interfaces, airflow
patterns, and the laboratory housing the system. These photographs
only hint at the realism of actual displays which provide depth cues
by showing each eye a view from a slightly different perspective and
also allowing the user to pan across the computer-generated
environment by turning his head. The user can select a menu option
with a word or a gesture, turn an airflow model to look at it from
another angle or reach out and "touch" the laboratory's walls and
desks.
    p.129
    The visual field of one eye, assuming the head is fixed, spans
180 degrees horizontally and 150 degrees vertically.....At viewing
distance [2 feet] each pixel subtends about two minutes of the
visual field, but the human eye can distinguish detail down to one
minute...a 20-inch-square-color monitor is available with
apporximately four million pixels in a 2000 by 2000 grid; viewed
from a distance of two feet each pixel subtends an angle of roughly
1.4 minutes...Motion paralax - the shift in background that occurs
when an observer looking at a point in space changes position - This
effect is achieved with the aid of a sensor the registers head
position and orientation.
    p.130
    Eye trackers bounce a beam of light off the cornea of the eye.
The direction in which the light is reflected indicates where the
user is looking: the point of regard...Trackers that attach to
eyeglasses can be had for a few thousand dollars, but they are not
very accurate...a pinpoint of infrared light onto the cornea and
detects its reflection with a wide-angle television camera placed
approximately 3 feet from the user. The camera stays locked on the

File ID: 00007

Subject: Human Factors
Key Word 2:   Eye tracking
Key Word 3:   Visual Cognition
Key Word 4:   Eye Resolution

Author: Foley, James D.
Title: Interfaces for Advanced Computing

-Media: Scientific American

Publication Date: 1-Oct-87
Publisher, City:
Page #:   127-135
Document ID:
Agency:
Type of Document: Magazine

Abstract/Citation:
eye in spite of considerable head movement unless the movement is
quite rapid...
    ...[explanation of DATAGLOVE]
    p.132
    techniques for the feeling or touching phenomena
    p.134
    see caption...The author found that users compare pipes that are
colored in and highlighted about 20 percent faster than they compare
outlined figures, but that further refinement does not significantly
improve comparison.
    p.135
    We found that users can compare colored-in images about 20
percent faster than outlined figures, but that the more
sophisticated shaded representations did not improve comparison time
any further.

File ID: 00008

Subject: Display Technology
Key Word 2:   Different Types
Key Word 3:   Pro/Cons
Key Word 4:

Author: Earl for Astronautics
Title: Display Technologies

Media: Lecture

Publication Date:
Publisher, City:
Page #:
Document ID:
Agency:
Type of Document: Lecture Notes

Abstract/Citation:
Pros and Cons of the following types of Displays

    Cathode Ray Tube (monochrome)
    Shadow Mask Color
    Color Beam Index
    Solid State Field Effect Distributed Cathode
    Flat Panel - Color Fixed Format
    Fixed Format Liquid Crystal
    Monochrome Active Matrix LCD
    Full Color Active Matrix LCD
    Distributed Cathode - Field Emmission Display

File ID: 00009

Subject: Human Factors
Key Word 2:   Interface
Key Word 3:
Key Word 4:

Author: Kocian, Dean   briefed by
Title: Visual World Subsystem

- Media: Super Cockpit Industry Days Agenda

Publication Date: 1-Apr-87
Publisher, City:
Page #:
Document ID:
Agency:
Type of Document: 2 day agenda/outline

Abstract/Citation:
in Visual World Subsystem section
    Human/Operator VCS interface persormance

    Display update time   <= 16.7 msec
    Position update time
        desired   <=33 msec
        unacceptable   <66 msec
    System Throughput (lag) Time
        desired   <=33 msec
        bothersome   35-50 msec
        objectionable   <50 msec
    Precision (resolution)   14 bits

File ID: 00010

Subject: Human Factors
Key Word 2:   Field of View
Key Word 3:   Visual Cognition
Key Word 4:

Author: Ellis, Stephen R.
Title: Visual Enhancements for Perspective Displays:
        Perspective Parameters
Media: International Conference on Systmes, Man and
        Cybernetics 1985
Publication Date: 1-Jan-85
Publisher, City: IEEE
Page #: 815-818
Document ID: CH2253-3/85/0000-0815
Agency: NASA Ames Research Center, Moffett Field, CA
Type of Document: Conference Proceedings

Abstract/Citation:
ABSTRACT:
    Experiments have been conducted examining systematic effects on
pursuit tracking of parameters defining perspective displays.
Targets on the displays were modeled to move in three dimensions and
were tracked by subjects using 2 two-axis joy-sticks. Some of the
patterns of tracking errors may be explainded by geometric
distortion due to projection of three-dimensional objects onto the
display surface. Time lags introduced by the users attempts to
interpret the three-dimensional position of the targets may also
contribute to tracking error.

    p.815
    Changing the FOV of the perspective projection while keeping the
distance to the reference point and the actual eye point constant
has two general effects on the resulitng image. 1) It causes a
magnifactioan or minification of the image depending upon whether
the FOV is decreased or increased respectively. 2) Additionally, it
causes a distortion in the projection of the image that arises
partly from the fact that the viewer's eye is not at the corret
center of the projection.
    p.816
    This distortion arises from ..."window assumption"in which the
viewer assumes the projectiors from the object to hes eye pass
streight through the picture plane.....The amount of distortion
depends upon 1) the ratio of the distance of the viewer's actual eye
point to the picture plane and the corresponding distance of the
center of the projection to the picture plane and 2) the height h'
of the image from the ecnter of the viewport on the display suface
(see fig) There is no distortion in the center of the image ....The
distortion is greatest at the edges of the projected picture,
because here the amount of bending of the projectors is greatest.

File ID: 00011

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:
Key Word 4:

Author: Yarbus, Alfred L.
Title: Eye Movements, and Vision

Media: Book

Publication Date: 1-Jan-67
Publisher, City: Plenum Press, New York, NY
Page #: 204-205
Document ID: Lib. Congress Card # 66-19932
Agency:
Type of Document: Book

Abstract/Citation:
p.204
    ...I pointed out earlier that even large saccades are often
involuntary. In many cases, saccades, sometime even a group or a
series of saccades apparently volutary in nature, are not entirely
under the observer's control. This "disobedience" is particularly
obvious in records of eye movements accompanying the perception of
optical illusions.

File ID: 00012

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Blind Spot
Key Word 4:

Author: Alpern, Mathew   ed. Hugh Davson
Title: The Eye

Media: Book

Publication Date: 1-Jan-67
Publisher, City: Academic Press, New York NY
Page #:   see below
Document ID:
Agency:
Type of Document: Book

Abstract/Citation:
    p.24
    Plotting the blind spot in the tertiary position of gaze,
originally described by Flick(1854).....Measurements on eight
subjects were very nearly identical.
    p.72-
    High Frequency Tremor - These are very fine oscillatory movements
of approximate median amplitude of 17.5 sec of arc with a range of
from just perceptible to 2 minutes of arc....[The frequency of tremor
was found to range from 30 to 100 Hz in different studies].. The
velocity of these eye movemnts at the maximum is about 20 min/sec.

    Slow drifts - 5 min of arc, duration less than .2 sec and
angular velocity of about 1 min of arc/sec.

    Rapid "Ficks" of Saccades - amplitude of 5.6 min of arc but can
range from 1 to 20 minutes (rarely larger than 10 minutes) Angular
velocity of 10 degrees/sec and are irregular in occurance.

    Irregular Movements - Although they cannot accurately be
described, they have an approximate peak-to-trough amplitude ranging
from 1-5 minutes of arc and freq of from 2-5 cycles/sec.

    p.73
    ...During the normal fixation, the instability of the oculomotor
system permits the eye to drift, and, consequently, the retinal
image drifts farther and farther away from some particular region of
the retina, it becomes more and more likely that a saccadic movement
will occur, tending to return the retinal image to that particular
region.
    p.76
    If the eye continued to look at the target for any length of time.
Prolonged fixation of a target under these condition resulted in
the disappearance of objects from the field of view. Thus it seems
that the involuntary eye movements of steady fixation are a
necessary feature for normal vision. This disappearance occurs

File ID: 00012

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Blind Spot
Key Word 4:

Author: Alpern, Mathew   ed. Hugh Davson
Title: The Eye

- Media: Book

Publication Date: 1-Jan-67
Publisher, City: Academic Press, New York NY
Page #:   see below
Document ID:
Agency:
Type of Document: Book

Abstract/Citation:
in about 5 sec.   It is greatest for low contrast images with minimal
amounts of motion... If a black bar 8' width and 2 degrees length is
seen, the black area soon appears to be invaded from the edges by
patches of light and is finally completely obliterated so that the
whole field appears uniformly illuminated...black bars tend to
become white rather than the white background to become darker.

    p.101
    ...Maximum velocity of about 21.42 degrees/sec for a lateral
fusional movement of 5.5 degrees, and this conpares rather well with
a maximum velocity of 20.2 deg/sec found by Alpern and Ellen for 6.6
deg lateral vergence movement which had accommodative but no fusional
components....The important point is that movements of this kind are
very much slower that saccadic eye movements.
    p.110

File ID: 00013

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Visual Cognition
Key Word 4:   Visual Feedback

Author: Gaarder, Kenneth R.
Title: Eye Movements, Vision and Behavior: A Hierarchical
        Visual Information Processing Model
Media: Book

Publication Date: 1-Jan-75
Publisher, City: Hemisphere Publishing Corp., Washington, D
Page #:
Document ID: ISBN 0-470-28895-7
Agency:
Type of Document: Book

Abstract/Citation:
    p. 24
    These studies showed that contrary to appearance at gross visual
inspection, the eye continues to move even during fixation.
    p.25
    The most important fact about fixation eye movements discovered
in this work was that the rapid jumps of the eye are still present,
even though the eye is apparently fixated. The reason the rapid
jumps cannot be seen is because they are too small. They are of a
size of about 10 minutes of arc (1/6 of a degree of arc on the globe
of the eye) or less, which is less than 0.01 inch of movment on the
surface of the eye and less than the diameter of 30 cone
receptors...
    p.31
    It was shown logically how the movement of the eyes, which
changes the way the visual stimulus impinges on the retinal image,
is a feedback linkage. Finally, it was shown experimentally that
changing the visual stimulus changes that pattern of eye movemtns,
which in turn must have the circular effect of changing the next
increment of new visual input.
    p.70
    ...even furing eye fixation there continues to be movemtn in the
form of slow drifts and a more rapic tremor.... Whereas the edge
information templates generated by eye jumps are produced by a
sudden displacement along a single vector within a time of 10 to 20
msec, the templates generated by drift occur more slowly over times
of from 20 to 100 msec and have a shifting vector....eye jumps and
drifts are of a size of about 5 minutes of arc, this means movement
may tranerse as many as 15 cone receptors....fine tremor, of about
20 to 30 sec of arc will traverse about 1 to 1.5 receptor.
    p.71
    ..eye almost never slowly slides over a scene. It is a worthshile
experiment to try to slowly slode your eyes across a room or across
a landscaps. Even if you achieved a feeling of success in doing so,
you would find if your eye movements were recorded or observed that
they were not smooth but were in fact a series of jumps.

File ID: 00013

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Visual Cognition
Key Word 4:   Visual Feedback

Author: Gaarder, Kenneth R.
Title: Eye Movements, Vision and Behavior: A Hierarchical
         Visual Information Processing Model
Media: Book

Publication Date: 1-Jan-75
Publisher, City: Hemisphere Publishing Corp., Washington, D
Page #:
Document ID: ISBN 0-470-28895-7
Agency:
Type of Document: Book

Abstract/Citation:
   p.73
   [see fig] Fixations 2 and 3 would be jumps to points of maximal
information is the scene. (The information of amximal because the
most numbers of edges and surgaces meet at these points.)...in
viewing a scene the eye tends to fixate on points of maximum
information.

   ...the density of retinal receptors and, therefore, the visual
acuity, drop off drastically as one moves away form the fovea to the
periphery of the retina

File ID: 00014

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Tracking Devices
Key Word 4:

Author: Tweed, Douglas
Title: Computing Three-Dimensional Eye Position Quaternions
         and Eye Velocity From Search Coil Signals
Media: Vision Research

Publication Date: 17-May-89
Publisher, City: U. of Western Ontario, London
Page #:
Document ID: 0042-6989
Agency:
Type of Document: Journal

Abstract/Citation:
Abstract - The four component rotational operators called
quaternions, which represent eye rotations in terms of their axes
and angles, have several advantages over other representations of
eye position (such as Fick coordinates): they provide easy
computations, symmetry, a simple form for Listings law, and useful
three -dimensional plots of eye movements. In this paper we present
algorithms for computing eye position quaternions and eye angular
velocity (not the derivative of position in three dimensions) from
two search coils (not necessarily orthogonal) on one eye in two or
three magnetic fields, and for locating primary position using
quaternions. We show how differentiation of eye position signals
yields poor estimates of all three comonents of eye velocity.

File ID: 00015

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Model
Key Word 4:

Author: Kinsner, W.
Title: An Eye Model For Ocular Pulse Analysis

Media: IEEE Engineering in Medicine & Biology Society

Publication Date: 1-Jan-89
Publisher, City: U. of Manitoba, Winnipeg
Page #:
Document ID: 2770-6/89/0000-0246
Agency:
Type of Document: Journal

Abstract/Citation:
References
    C. C. Mow,"A theoretical model of the cornea for use in studies
of tonometry," Bulliten of Mathematical Biophysics, vol. 28, pp.
585-634, 1966
    F. H. Adler, Physiology of the Eye. Saint Louis: The C. V. Mosby
Co., 4th Ed., 1965

File ID: 00016

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Visual Imagery
Key Word 4:

Author: Brandt, Stephan A.
Title: Experimental Evidence for Scannpath Eye Movements
        During Visual Imagery
-Media: IEEE Engineering in Medicine & Biology Society

Publication Date: 1-Jan-89
Publisher, City: U. of Manitoba, Winnipeg
Page #:
Document ID: 2770-6/89/0000-0246
Agency:
Type of Document: Journal

Abstract/Citation:
   The equipment used for recording EMs consisted of a video-based
EM tracking system and a software package for the aquisition,
analysis and plotting of eye movements.
   [3] Zikmund V., "Oculomotor Activity During Visual Imagery of a
Moving Stimulus Pattern", Studia Psychologica VIII:254-273 (1966).
   [4] Marks D., "Visual Imagery Differences and Eye Movements in
the Recall of Pictures", Perception & Psychophys., 14:407-412 (1973).
   [7] Hacisalihzade S. et al., "Automatic Analysis of Eye Movements
with a PC", 11th Ann. Int'l. Conf. IEEE/EMB Soc., Seattle, (1989).
   [8] Sheehan P.W. et al., "Some Variables Affecting the Vividness
of Imagery in Recall", Br. J. Psychol., 60:71-80, (1969).

File ID: 00017

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Field of View
Key Word 4:

Author:
Title: Comparison of Visual Fields

-Media:

Publication Date:
Publisher, City:
Page #:
Document ID:
Agency:
Type of Document: Individual Graphs

Abstract/Citation:
3-D Graphical Analysis of Eye IMFOV

File ID: 00018

Subject: Human Factors
Key Word 2:   Stereoscopic Fusion
Key Word 3:
Key Word 4:

Author: Yeh, Yei-Yu
Title: Limits of Fusion and Depth Judgment in Stereoscopic
        Color Displays
-Media: Human Factors

Publication Date: 1-Jan-90
Publisher, City: Honeywell, Inc., Phoenix
Page #:   46-60
Document ID:
Agency:
Type of Document:

Abstract/Citation:
    p. 46
    Subjects could perform a parallel search for a red taget in one
depth plane without interferencefrom red items in the opther depth
plane, but they could not filter out red items moving in the
opposite direction at the same depth.
    They visual search performance appears to depend on the distance
from noise or distractor elements to the observer as well as the
distance from distractor elements to the target.
    p. 48
    The limits were specified by the sensory limitations in fusing
stereo images and were measured by the diplopia threshold-the
threshold between singleness of vision and double vision.
  Considering that the limits of fusion vary with many factors (see
Arditi, 1987, for a thorough review) and a wide range of diplopia
thresholds have been found in the literature (Mitchell, 1966b;
Duwaer and Van Den Brink, 1981), it is crucial to empirically define
the limits of disparity that can be fused with operationally
realistic stimulus parameters on an electronic stereoscopic display.
    p. 49
    Because cros talk between the left- and right-eye images
essentially results in "binocular visual noise," it might be
expected to affect fusion and/or vergence responces. Thus in the
first experiment the effect of interocular cross talk on the ability
to fuse stereo images was investigated by manipulating 2 display
variables: stimulus color and vertical screen position. Three
factors determine the amount of interocular cross talk between the
two eye views: (1) the amount of light transmitted by the LC shutter
in its off or closed state; (2) phosphor persistence; and (3)
vertical screen position of the images.
    p. 50-52
    Fusion limits (i.e., diplopia thesholds) for a 66.04-cm viewing
distance.
    Four independent variables were entered into the analysis of
fusion data: group variable (order of color presentation), 3 within
-subjects variables (stimulus color, vertical screen position, and

File ID: 00018

Subject: Human Factors
Key Word 2:   Stereoscopic Fusion
Key Word 3:
Key Word 4:

Author: Yeh, Yei-Yu
Title: Limits of Fusion and Depth Judgment in Stereoscopic
        Color Displays
-Media: Human Factors

Publication Date: 1-Jan-90
Publisher, City: Honeywell, Inc., Phoenix
Page #:   46-60
Document ID:
Agency:
Type of Document:

Abstract/Citation:
direction of disparity). Table 2 provides the basic numerical
results on the limits of fusion as a function of stimulus color,
vertical screen position, and stimulus duration. Vertical screen
position has a significant effect on the mean diplopia threshold.
 Average fusion limits were significantly greater at the top screen
position than at the bottom position. Fusion limits at the center
position were not significantly different from limits at either the
top or bottom. The interaction between stimulus color and vertical
position was significant. For white images, fusion limits at the
top position were significantly greater than limits at the bottom
and not significantly different from limits at the center position.
     Limits of fusion for field-sequential stereoscopic displays and
how interocular cross talk may affect these limits.
     Thus the mean diplopia thresholds found for the 200-ms exposure
duration in the present experiment-approximately 27 min arc for
crossed disparity and 24 min arc for uncrossed disparity-are in
reasonable accord with the basic vision literature on limits of
fusion. Resulting in mean diplopia thesholds of about 4.93 deg for
crossed disparity and 1.57 deg for uncrossed disparity. Thus a more
realistic range of between 27 min arc crossed disparity to 24 min
arc uncrossed disparity is recommended for coding symbols subtending
a visual angle.
     p. 55
     Whereas ratings of eyestrain and headaches engendered by viewing
the stereoscopic images were not significantly affected by stimulus
color, there was a trend toward a higher incidence of reported
eyestrain and headaches with the white test stimulus, suggesting
that interocular cross talk may influence the development of
asthenopia.
     p. 59 -references
     Arditi, A. (1987). Binocular vision. In K. R. Boff, L, Kaufman,
and J. P. Thomas (Eds.), "Handbook of perception and human
performance (pp. 23-1-23-41). New York: Wiley.
     Duwaer, A. L., and Van Den Brink, G. (1981). What is the diplopia
threshold? "Perception and Psychophysics", 29, 295-309.

File ID: 00018

Subject: Human Factors
Key Word 2:   Stereoscopic Fusion
Key Word 3:
Key Word 4:

Author: Yeh, Yei-Yu
Title: Limits of Fusion and Depth Judgment in Stereoscopic
         Color Displays
Media: Human Factors

Publication Date: 1-Jan-90
Publisher, City: Honeywell, Inc., Phoenix
Page #:   46-60
Document ID:
Agency:
Type of Document:

Abstract/Citation:
    Mitchell, D. E. (1966b). A review of the concept of "Panum's
fusional areas." "American Journal of Optometry", 43, 387-401.

File ID: 00020

Subject: Human Factors
Key Word 2:   Eye Sensitivity
Key Word 3:
Key Word 4:

Author: Ripps, H. and R.A. Weale
Title: The eye

-Media: Book

Publication Date: 1-Jan-76
Publisher, City: Academic Press, New York
Page #: 55-195
Document ID:
Agency:
Type of Document:

Abstract/Citation:
   p.71
   Fig. 2.16 Comparison of relative foveal sensitivites measured with
different criteria as a function of the wavelenth in nm.

   p.81
   Fig. 2.24 Correlation of rod counts (Qin thousands per sq mm) with
relative reciprocal of threshold and relative pigment dentsity
differences as determined by fundus reflectometry.

   p.89
   Table III Absolute thresholds for human rods and cones. The
values in brackets relate to quanta at the retina and were obtained
on the assumption that 4% of the light incident on the cornea is
reflected 50% lost in the pre retinal media and that cones absorb
maximally 40% of the light that reaches them.
   p.165
   lightrays passing through the pupillary periphery when D is
greater than 3 mm are visually less efficacious then those passing
through the center because they strike the cones at progressively
larger angles
   compensate for the increase in retinal illlumination which itself
improves visual acuity also if the pupillary diameter is kept
constant

   p. 166
   Fig. 4.22 Visual acuity as a function of log reference luminance
for various pupil diameters. The reference luminance values
indicated are those of a reference field viewed by the other eye
through a 2 mm pupil diameter. The pupil diameters in millimeters
are indicated at the right of the curves. The curves are emperical.

   p. 195
   There is no suggestion that the sensation is such as to enable one
to count the flaches, which may number as many as 60/s before fusion
occurs. Angel et.al. mention that the fusion frequency measured

File ID: 00020

Subject: Human Factors
Key Word 2:   Eye Sensitivity
Key Word 3:
Key Word 4:

Author: Ripps, H. and R.A. Weale
Title: The eye

-Media: Book

Publication Date: 1-Jan-76
Publisher, City: Academic Press, New York
Page #:   55-195
Document ID:
Agency:
Type of Document:

Abstract/Citation:
near the foveal threshold is 11 c/s, that for rods 8 c/s

File ID: 00021

Subject: Human Factors
Key Word 2:   Eye Resolution
Key Word 3:   Fusion Frequency
Key Word 4:

Author: Vincent, John David
Title: Fundamentals of Infrared Detector Operation and
         Testing
-Media: book

Publication Date: 1-Jan-90
Publisher, City: John Wiley & Sons, New York
Page #: 393-397
Document ID: ISBN 0-471-50272-3
Agency:
Type of Document:

Abstract/Citation:
   p. 395
   the cones have a diameter of about 1.5 um. Since the image
distance is 17mm, the cones subtend an angle of about0.1x10**-3 rad,
or about .3 min of arc
   if we use an 8mm iris diameter and a wavelength of .55um, we
obtain a diffraction-limited angle of .17 mrad.

   p. 397
   Based on the analysis, we would predict the resolution of the eye
to be a little under 1 minute of arc; this agrees well with measured
values: Alpern states that in the construction of his eye charts,
the Dutch opthamologist Snellen used 1 minute of arc as the minimum
separable angle under optimum condition. Winter (1971) indicates
that the probability of detecting a target rises to 50% as the
target subtends an angle of about .23 nrad, .8 minute of arc.

File ID: 00022

Subject: Human Factors
Key Word 2:   Fusion Frequency
Key Word 3:
Key Word 4:

Author: Waldvogel, Jerry A.
Title: The Bird's Eye View

-Media: American Scientist, Volume 78

Publication Date: 7-Jan-90
Publisher, City:
Page #:   342-353
Document ID:
Agency:
Type of Document: article

Abstract/Citation:
   p. 347
   retinal habituation, the tendency of neural cells to become less
sensitive when there is continual sensory input
   Acuity, or the ability to distinguish fine detail, is determined
by two factor: the overall shape of the eye and lens, and the
density of cell in the retina

   p. 349
   flicker-fusion frequency only at frequencies lower than about 60
hz can people resolve sequential moving images. At frequencies
higher than 60 hz the images blend together, creating the illusion
of continuous movement on which the motion picutre depends. At
still higher frequencies say 80 hz, our interpretation of motion is
more erratic, because the image does not remain on any one receptive
field long enough to register.

File ID: 00023

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Bour, Lo J.
Title: The Double Magnetic Induction Method for Measureing
        Eye Movement-Results in Monkey and Man
‾ Media: IEEE transactions on Biomedical Engineering vol BME-
        31 no.5
Publication Date: 1-May-84
Publisher, City: IEEE
Page #:   419-427
Document ID:
Agency:
Type of Document: paper

Abstract/Citation:
Abstract - We describe an improved version of the double magnetic
induction method for measuring eye movement, proposed by Reulen and
Bakker. The idea behind the double magnetic method is to detect eye
position indirectly by determining the strength of the induced
secondary magnetic field of a short-circuited coil on the subject's
eye caused by a primary magnetic field. A signal related to eye
position is obtained from a detection coil, placed in front of the
eye, without the need of connecting wires.
    Instead of the short-circuited Collewijn-coil, we use a polished
metal ring on the sclera of the eye. This makes the method more
comfortable for the subject and results in a larger signal
amplitude. As a further improvement, the signal of the detection
coil, consisting of a primary induced component and a relatively
weak secondary component, is differentially amplified together with
the signal of a compensation coil, consisting of only a primary
component, to avoid instrument overload and noise. We have used the
method successfully in both man and monkey. Technical
specifiactions of the method, as well as a procedure to correct for
its inherent nonlinearity, are described in detail.

File ID: 00024

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Collewijn, H.
Title: Precise Recording of Human Eye Movements

Media: Vision Research

Publication Date: 1-Jan-75
Publisher, City: Pergamon Press, Great Britian
Page #:   447-449
Document ID:
Agency:
Type of Document: paper

Abstract/Citation:
The precise recording of human eye movements is often of high
interest in fundamental and clinical investigations of the
oculomotor and visual systems.
    An ideal method should satisfy the following requirements:
  (1) sufficient resolution, linearity and dynamic range (in time
and space)
    (2) flexible sensitivity level for different applications
    (3) good stability
    (4) no interference with normal vision
    (5) relative insensitivity to translational head movements; no
need for rigid head fixation
    (6) simultaneous measurement of horizontal, vertical and torsional
movements
    (7) insensitivity to illumination conditions, closure of eyelids,
electromyographical and other electrical interference
    (8) easy, non-traumatic applicability to subjects without proir
experience or extraordinary motivation

File ID: 00025

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Collewijn, H.
Title: Human ocular counterroll: assessment of static and
        dynamic properties from electromagnetic scleral coil
Media: Experimental Brain Research

Publication Date: 1-Jan-85
Publisher, City: Springer-Verlag
Page #: 185-196
Document ID:
Agency:
Type of Document: paper

Abstract/Citation:
Static and dynamic components of ocular counterroll as well as
cyclorotatory optokinnetic nystagmus were measured with a scleral
search coil technique. Static counterroll compensated for about 10%
of head roll when the head was tilted to steady positions up to 20
deg from the upright position. The dynamic component of counterrollm
which occurs only whil the head is moving, is much larger.   It
consists of smooth compensatory cyclorotation opposite to the head
rotation, interrupted frequently by ssaccades moving in the same
direction as the head.   During voluntary sinusoidal head roll,
cyclorotation compensated from 40% to more than 70% of the head
motion.   In the range 0.16 to 1.33 Hz, gain increased with frequency
and with the amount of visual information. The lowest values were
found in darkness.   The gain increased in the presence of a visual
fixation point and a further rise was induced by a structured visual
patter. Resetting saccades were made more frequently in the dark
than in the light. These saccades were somewhat slower than typical
horizontal saccades. Cyclorotatory optokinetic nystagmus could be
induced by a patterned disk rotating around the visual axis. It was
highly variable even within a same subject and had in general a very
low gain (mean value about 0.03 deg/s). It is concluded that
cyclorotational slip velocity on the retina is considerable reduced
by counterroll during roll of the head, although the residual
cyclorotation after the head has reached a steady position is very
small.

File ID: 00026

Subject: Human Factors
Key Word 2:   Eye Tracking
Key Word 3:   Perception
Key Word 4:

Author: Tello, Ernest R.
Title: Between Man and Machine

Media: BYTE

Publication Date: 1-Sep-88
Publisher, City:
Page #: 288-293
Document ID:
Agency:
Type of Document: article

Abstract/Citation:
   p. 290
   Basically, eyetracking works by using special cameras that lock
onto the eye, then bounce a beam off its cornea and continually
record the direction in which the light is reflected. Current
techniques allos resolution fine enough to determine which word on a
computer screen you're focusing on.
   p. 291
   Several eye-tracking projects are currently uner way by both
universities and commercial firms. At Texas A&M University, there
has been ongoing research on an eye-tracking system using the frames
of eyeglasses equipped with infrared-emitting diodes and
phototransistors. Basically, this device uses electro-optical
sensing in such a eay that the positioning of the eye determines the
amount and intersity of the light emitted to the phototransistors.
   p. 292
   The Virtual Interactive Environment Workstation (VIEWS) uses
head-tracked and head-mounted displays, gesture-tracking, and voice
input and output. (NASA's Lanfley research branch is also working
on head-mounted VIEWS devices.)
   ....The system uses various video mixing and switching devices,
combining both commercially available and custom equipment to merge
inages from a varity of sources into a single integrated
representation.
   VIEWS also incorpoertes a commerciall available binaural,
three-dimensional sound system. With headphones, you can hear
sounds, either from withing or outsode your immediate field of view.
A speech-recognition module lets you give vocal commands in a
natural, conversational manner.
   NASA's interest in these technologies stems form its desire to
incorporate on the planned space station an implementation of the
concept of "telepresence" - the ability to put yourself into almost
any environment you can envision. The agency has committed itself to
using the most advanced user-interface technology for control of
autonomous and semi-autonomous telerobotics devices.

File ID: 00027

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Ellis, Stephen R.
Title: A New Illusion of Projected Three-Dimensional Space

Media: NASA Technical Memorandum 100006

Publication Date: 1-Jul-87
Publisher, City: NASA , Moffett Field, California
Page #:
Document ID: 100006
Agency: NASA
Type of Document:

Abstract/Citation:
    ...use knowledge of the typical shape or size of an object to estimate its depicted orientation.
    This kind of familiarity cue to picture interpretation does not necessarily require a complete description of the pictured object. Certain features of the object, such as parallelism or perpendicularity of some of its parts, may provide much of the information needed to calculate the depicted prientation of an object from its picture. (Grunwald and Ellis, 1986) ....It has a powerful influence on the spatial interpretation of not only pictures, but also of stereoscopically viewed, 3d objects.
    Thus, though familiarity cues can assist interpretation of an image, the effects of the projective ambiguities persist and projected images often have multiple interpretations.
    .....A key set of assumptions that must be implicitly made concerns the relative position in depth of the curve where it appears to cross itself, i.e. the front/bask assignment. If both crossing lines are assumed to be at the same depth when they cross, the entire curve may seem to be confined to a plane in space.

File ID: 00029

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: Smith, David M.
Title: A Methodology For Developing Cooperating Expert
        Systems
Media:

Publication Date: 1-Jan-87
Publisher, City: Lockheed-Georgia Co.
Page #:
Document ID:
Agency: Lockheed, US Air Force
Type of Document: technical paper

Abstract/Citation:
   Pilot's Associate (PA)
   ..five distinct areas
   ..a sixth module, the Mission Manager
   ..this paper explores the methodology for defining the functional
requirements and interfaces between the cooperating expert systems.
   Finally, it describes the progress and experience to date...

File ID: 00030

Subject: Artificial Intel.
Key Word 2:   Interface
Key Word 3:   Control
Key Word 4:

Author: Broadwell, Martin
Title: Pilot's Associate: Vehicle For AI Application

Media:

Publication Date: 1-Jan-86
Publisher, City: Lockheed-Georgia, Marietta
Page #:
Document ID:
Agency: Lockheed
Type of Document: technical paper

Abstract/Citation:
   Pilot's Associate (PA)
   ..PA provides an ideal vehicle for generating the requirements
for, developing, demonstrating, and evaluating the application of
machine intelligence technologies to critical defence problems.
   ..implimentation of expert systems
   ..coordinate automated components

File ID: 00031

Subject: Artificial Intel.
Key Word 2: Interface
Key Word 3: Control
Key Word 4:

Author: Aeronautical Sytems Division
Title: The Pilot's Associate

-Media: United States Air Force

Publication Date: 1-Feb-86
Publisher, City: Wright-Patterson AFB, OH
Page #: 2
Document ID: PAM #86-039
Agency: US Air Force
Type of Document: Fact Sheet

Abstract/Citation:
   ...the aircraft itself must communicate to the pilot what is important, and do it in a manner which is both timely and easily understood.
   ...five function areas: systems status, mission planning, situation assessment, tactics, pilot-vehicle interface.

File ID: 00032

Subject: Artificial Intel.
Key Word 2:   Integrated
Key Word 3:   Expert Systems
Key Word 4:

Author: Rao, Ming
Title: Intelligent Control Workstation

-Media: IEEE

Publication Date: 1-Jan-89
Publisher, City: Rutgers, Piscataway
Page #: 1
Document ID: CH2759-9/89/0000-1011
Agency: Department of Chem. and Bio. Engineering
Type of Document: technical paper

Abstract/Citation:
   This paper describes an intelligent control workstation that can
be used to design various intelligent control systems for the
real-time applications. The software library accessible by the
workstation consists of several numeric simulation packages and
expert systems, which cover a wide spectrum of control knowledge.

File ID: 00033

Subject: Artificial Intel.
Key Word 2:   Integrated
Key Word 3:   Expert Systems
Key Word 4:

Author: McGraw, Karen L.
Title: Integrated Systems Development

_Media:

Publication Date: 1-Dec-86
Publisher, City:
Page #:   1,2,4
Document ID:
Agency: Texas Instruments
Type of Document: paper

Abstract/Citation:
    ...AI because it shows promise as a methodology for solving
problems characterized by incomplete data, information overload, and
dynamically changing environments.
    ...improve the effectiveness of future military systems, while
decreasing the workload of humans in stressful and potentially
dangerous situations. Its broadcast objective is to develop machine
"intelligence" that permits the development of systems that closely
assist human operators.
    Conventional programming methods...AI programming
methods...traditional computing uses algorithmic methods.
    Examples of appropriate emergencies include: loss of canopy,
hydraulic system failure...

File ID: 00034

Subject: Artificial Intel.
Key Word 2:   Real World
Key Word 3:
Key Word 4:

Author: Guffey, James A.
Title: Real World Artificial Intelligence: Make a
         Realistic Assessment
- Media:

Publication Date: 7-May-87
Publisher, City: McDonnell Douglas, St. Louis
Page #: 1
Document ID: MCAIR NO. 87-004
Agency: McDonnell Douglas
Type of Document: technical paper

Abstract/Citation:
   AI...no doubt that substantial gains can be made by exploiting AI
for small systems and specific problem domains...

File ID: 00035

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: Keller, Kirby J.
Title: An Expert Planning System for Air-to-Air Combat
         Management
-Media:

Publication Date: 7-Nov-84
Publisher, City: McDonnell Douglas, St. Louis
Page #:   1
Document ID: MCAIR NO. 84-025
Agency: McDonnell Douglas
Type of Document: technical paper

Abstract/Citation:
   ...planning systems are being applied to real-world problems
characterized by dynamic domains with mathematically complex
representations, uncertainty of current and future domain states,
and unresponsiveness to the planner's actions.
   In this application, the planning system combines expert fighter
pilot heuristics with detailed mathematical functions that evaluate
and execute these heuristics.

File ID: 00036

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: McCoy, Michael S.
Title: Human Performance Models Applied To Intelligent
         Decision Support Systems
Media:

Publication Date: 22-May-87
Publisher, City: McDonnell Douglas, St. Louis
Page #:   1-3
Document ID: MCAIR NO. 87-005
Agency: McDonnell Douglas
Type of Document: technical paper

Abstract/Citation:
   Human Performance Model...model will be one element of a decision
and task support system. It will predict the operator's forthcoming
tasks, anticipate upcoming decisions, and formulate any necessary
decision or execution aids. Second...
   "Human Performance Model" implies a concentration on the human's
role in a man-machine system...
   When a new demandis generated, or an existing demand is
eliminated, the situation must be reassessed...a queue of demands
exist...focusing on high priority demands for consideration...using
a heuristic process of elimination.
   A list of candidate task/procedures...

File ID: 00037

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: Gearhart, Larry M.
Title: Expert Systems: Technology and Applications---
          Managing the Development and Deployment of Expert
Media:

Publication Date: 9-Jan-89
Publisher, City: IEEE
Page #:   1
Document ID: CH2759-9/89/0000-1004
Agency: TRW Military Electronics and Avionics Division
Type of Document: technical paper

Abstract/Citation:
   The logic paths in an expert system are nonlinear...
   ...to compensate for that complexity. This paper examines the E&V
problem within the context of the life cycle.

File ID: 00038

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: Martz, Steve
Title: Expert System-Conventional Processing Interface

‾Media:

Publication Date: 9-Jan-89
Publisher, City: IEEE
Page #: 1
Document ID: CH2759-9/89/0000-1020
Agency: Boeing Military Airplanes
Type of Document: technical paper

Abstract/Citation:
  The evolution of an expert system...involved...prototype on an AI
workstation...
   ...features which are superfluous in the delivery environment.
 Such features, which free the system developer from mundane,
time-consuming, steps during prototype development...
   ...development of a cockpit information manager

File ID: 00039

Subject: Artificial Intel.
Key Word 2:   Expert Systems
Key Word 3:
Key Word 4:

Author: Cowin, Richard
Title: Real-Time Adaptive Control of Knowledge Based
        Avionics Tasks
Media:

Publication Date: 9-Jan-89
Publisher, City: IEEE
Page #:   1
Document ID: CH2759-9/89/0000-1175
Agency: Northrop Corp.
Type of Document: technical paper

Abstract/Citation:
   Advanced decidion making capabilities...
     ...distributed, fault-tolerant, software architecture that
permits the real-time prioritization and sceduling of these tasks.

File ID: 00043

Author: Jane Mulligan, Alan K. Mackworth, Peter Lawrence
Title: A Model-based Vision System for Manipulator Position
          Sensing
Media: IEEE

Abstract/Citation:
Abstract

    The task and design requirements for a vision system for
manipulator position sensing in a telerobtic system are described.
Model-based analysis-by-synthesis techniques offer generally
applicable methods with the potential to meet the system's
requirement for accurate, fast, and reliable results. Edge-based
champfer matching allows efficient computation of a measure, E, of
the local difference between the real image and a synthetic image
generated from arm and camera models. Gradient descent techniques
are used to minimize E by adjusting joint angles. The dependence of
each link position on the position of the link preceding it allows
the search to be broken down into lower dimensional problems.
Intensive exploitation of geometric constraints on the possible
position and orientation of manipulator components results in a
correct and efficient solution to the problem. Experimental results
demonstrate the use of the implemented prototype system to locate
the boom, stick and bucket of an excavator, given a single video
image.

File ID: 00044

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:   Monoscopic
Key Word 4:

Author: Yeh, Yei-Yu
Title: Visual Performance with Monoscopic and Stereoscopic
        Presentations of Identical Three Dimensional Visual
⌐Media: Society for Information Display 90 Digest

Publication Date: 1-Jan-90
Publisher, City:
Page #:   359-361
Document ID: ISSN0097-0966X/90/0000-359
Agency: Society for Information Display
Type of Document:

Abstract/Citation:
   p.359   The advantages of stereoscopic displays are linked to
their ability to simulate three-dimensional (3-D) objects or visual
scenes with a degree of fidelity higher that either standard
two-dimensional (2-D) displays or 2-D displays supplemented with
perspective graphics (2.5-D).   Stereoscopic cues can enhance visual
perception and performance.....constrast sensitivity and form
recognition can be improved by adding stereopsis to monocular visual
cues.1 and observations from anatomical and physiological studies
have revealed the existence of an independent neural pathway which
processes stereopsis in parallel to the pathway for processing color
and form.2   The benefits of incorporating stereopsis in electronic
displays have been demonstrated .....: enhanced fore-aft distinction
of a wire-frame globe3, improved positioning of a cursor within a
3-D display presentation4, and facilitated accuracy in counting the
number of target aircraft in a spacial quadrant of an air-to-air
situation display5..... Kim et. al. showed that stereopsis improved
the performance of tracking a three-dimensional moving input target
point when the visual scene was projected through a 0 degree or a 90
degree perspective orientation. With a 45-degree viewing
perspective, however, stereopsis did not provide additional
performance benefits when vertical lines were used to reference
target and cursor dots to the ground plane. The objective of the
experiment described in this papeer was to directly compare visual
performance with a perspective display format presented both
monoscopicaly and stereoscopically with the same visual display.
 The visual task involoved speeded judgements of the relative
spatial position between two geometric test targets embedded within
a schematic representation of a perspective ground plane with
reference objects.
   METHOD   All visual stimuli were displayed with a Tektronix
SGS-430 Stereoscopic Color Monitor coupled to a modified set of
Tektronix 808-012 Four Segmented Liquid Crystal Stereo Goggles,
which served as the field-sequential stereoscopic shutter
device.....The effective display addressability was   512 x 512
pixels in both stereoscopic and monoscopic modes. One pixel

File ID: 00044

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:   Monoscopic
Key Word 4:

Author: Yeh, Yei-Yu
Title: Visual Performance with Monoscopic and Stereoscopic
        Presentations of Identical Three Dimensional Visual
Media: Society for Information Display 90 Digest

Publication Date: 1-Jan-90
Publisher, City:
Page #: 359-361
Document ID: ISSN0097-0966X/90/0000-359
Agency: Society for Information Display
Type of Document:

Abstract/Citation:
subtended a visual angle of approximately 2.2 min arc at the test
viewing distance of 26 inches.....   p.360   RESULTS   The most
important findings were the performance benefits provided by the
addition of stereopsis over the monoscopic display presentations.
  Mean reaction times were significaly faster with the stereoscopic
than with the monoscopic perspective presentation in five of the six
experimental conditions. The only condition where stereopsis did
not shorten reaction time was when subjects could easily use the
ground reference plane as a grid to judge the relative distance
between test targets under the 90-degree viewing orientation.
  Facilitation of reaction time was estimated to range from 22
percent (altitude judgments for targets viewed from the 15 degree
orientation) to 41 percent (distance judgements for targets viewed
from the 15 degree orientation). .....   The resutls also revealed
that judgments of spatial relations were strongle influenced by the
position of test targets within the perspective reference volume.
  As indicated in Figures 5 and 6 when test targets were closer to
the viewing reference point(i.e. in front of the midpoint of the
ground reference plane), monocular depth cues were more effective
and judgments of relative spatial relations were faster and more
accurate than when targets were positioned further away from the
view point. Thus, as monocular depth cues become less salient or
ambiguous for targets further away from the view point, the relative
visual preformance benefits of stereoscopic presentation become
progressively larger.   p.361   CONCLUSIONS   The present experiment
directly compared visual performance with a perspective display
format presented both monoscopically and stereoscopically with the
same visual display. The most important findings were the
performance benefits provided by the addition of stereopsis over the
monoscopic display presentations. Mean reaction times were faster
and spatial judgments were more accurate with the stereoscopic than
with the monoscopic perspective presentations......stereopsis was
found to facilitate the repidity of spatial judgments by as much as
41 percent under some conditions.
        ....the results also revealed the conditions and constraints

File ID: 00044

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:   Monoscopic
Key Word 4:

Author: Yeh, Yei-Yu
Title: Visual Performance with Monoscopic and Stereoscopic
         Presentations of Identical Three Dimensional Visual
Media: Society for Information Display 90 Digest

Publication Date: 1-Jan-90
Publisher, City:
Page #: 359-361
Document ID: ISSN0097-0966X/90/0000-359
Agency: Society for Information Display
Type of Document:

Abstract/Citation:
under with stereopsis can be expected to improve spatial judgements.
 When spatial judgments could be easily executed by using reference
information available from monocular depth cues in the displayed
image, stereopsis did not provide significant performance benefits.
 However, under conditions where monocular depth cues became less
salient or ambiguous, stereopsis enable observers to resolve and
perceptual ambiguities and maintain a relatively high level of
visual performance.

File ID: 00045

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Ponce, Jean
Title: On Recognizing and Positioning Curved 3D Objects
          form Image Contours
Media: Workshop on Interpretation of 3D Scenes Proceedings

Publication Date: 1-Nov-89
Publisher, City: Computer Society Press
Page #:   61-67
Document ID: ISBN 0-8186-2007-2
Agency: IEEE
Type of Document: Proceedings

Abstract/Citation:
ABSTRACT
   A new approach is presented for explicitly relating the shape of
image contours to models of curved three-dimensional objects. This
relationship is used for object recognition and positioning.
   Object models consist of collections of parametric surface patches
and their intersection curves; this includes nearly all
representations used in computer aided geometric design and conputer
vision. The image contours considered are the projection of surface
discontinuities and occluding contours. Elimination theory provides
a method for constructing the implicit equation of the image
contours of a object observed under orthographic or perspective
projection. This equation is parameterized by the object's position
and orientation with respect to the observer. Determining these
parameters is reduced to a fitting problem between the theoretical
contour and the observed data points. The proposed approach readily
extends to parameterized models. It has been implemented for a
simple world composed of various surfaces of revolution and
successfully tested on several real images.

File ID: 00046

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:   Feedback
Key Word 4:

Author: Allen, Peter K.
Title: Acquisition and Interpretation of 3-D Sensor Data
          From Touch
- Media: Workshop on Interpretation of 3D Scenes Proceedings

Publication Date: 1-Nov-89
Publisher, City: Computer Society Press
Page #:   33-40
Document ID: ISBN 0-8186-2007-2
Agency: IEEE
Type of Document: Proceedings

Abstract/Citation:
ABSTRACT
   Acquisition of 3D scene information has focused on either passive
2D imaging methods (stereopsus, structure from motion etc.) or 3D
range sensing methods (structured lighting, laser scanning etc.)
Little work has been done in using active touch sensing with a
multi-fingered robotic hand to acquire scene descriptions, even
though it is a well-developed human capablilty. Future robotic
systems will need to use dextrous robotic hands for tasks such as
grasping, manipulation, assembly, inspection and object reconition.
   This paper describes our use of touch sensing as part of a larger
system we are building for 3D shape recovery and object recognition
using touch and vision methods. It focuses on three exploratory
procedures we have built to acquire and interpret sparse 3D touch
data: grasping by containment, planar surface exploration and
surface contour exploration. Experimental results for each of these
procedures are presented.

Abstract/Citation:
ABSTRACT
   We present a Stereo Vision System in which we attempt to achieve
robustness with respect to scene characteristics, from textured
outdoors scenes to environments composed of highly regular man-made
objects. It offers the advantages of both area-based (dense-map)
and feature-based (acurrate disparity) processing by combining them
whenever possile. We are able to generate a disparity map that is
sufficiently accurate to allow us to detect depth and surface
orientation discontinuities, provided that the resolution is fine
enough. We use an area-based cross-correlation along with an
ordering constraint and a weak surface smoothness assumption to
produce an initial disparity map. Unlike other approaches, however,
a match is accepted only if both views agree on a correlation peak
and this peak is strong enough. This disparity map is a blurred
version of the true one, however, because of the smoothing inherent
in the correlation. The problem is most acute at Co (depth) and C1
(crease) discontinuities, but can be mitigated by introducing the
edge information: the disparity at edgels is fixed. It is important
to note that this method gives an active role to the edgels parallel
to the epipolar lines, which are discarded in most feature-based
systems. We have obtained very good results on complex scenes in
different domains and have been able to locate visable surfaces,
penumbral and off-the-edge areas, and depth and in some cases
orientation discontinuities in the images.

File ID: 00048

Abstract/Citation:
ABSTRACT
  This paper is concerned with the problem of surface reconstruction
from stereo images for large sceen having large depth ranges, where
it is necessary to aim cameras in different direction and to fixate
at different objects. This paper presents an approach to acquiring
coarse structural information about the scene in the vicinity of the
next fixation point during the current fixation, and utilizing this
information for surface reconstruction in the vicinity of the next
fixation point. The approach invloves processing of peripheral, low
resolution parts of the current images away form the image center,
in addition to accurate surface estimation from the central, high
resolution parts containting the fixated object. The processing of
the low resolution parts yields coarse surface estimates to be
refined after the cameras have refixated, and the parts of the scene
around the new fixation point (currently at low resolution) are
imaged more sharply. The coarse estimates are obtained from both
stereo and focus.   The choice as to which estimate is actually used
depends on which one is determined to be more accurate in the give
situation. Thus, the approach presented also involoves dynamic
integration of the use of stereo and focus as sources of depth
information, in addition to integrating multiresolution image
acquision and their coarse-to-fine processing.

Abstract/Citation:
ABSTRACT
   This paper is concerned with the recovery of three-dimensional
scene geometry from binocular stereo disparity. In order to
accomplish this goal an analysis of disparity is presented. The
analysis makes explicit the geometric relations between a stereo
disparity field and a differentially projected scene. These
results show how it is possible to recover scene properties, such as
relative surface depth and orientation, in a direct fashion from
stereo disparity. As a particualr application of the analysis, a
method is presented for recovering the discontinuities of surfaces
from stereo disparity. The results of applying this method to both
natural and synthetic binocular stereo disparity information also
are presented.

File ID: 00050

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:
Key Word 4:

Author: Balasubramanyam, Poornima
Title: Early Identification of Occulsion In Stereo-Motion
         Image Sequences
-Media: Image Understanding Workshop

Publication Date: 1-Jan-89
Publisher, City: Morgan Kaufmann Publishers Inc., San Mateo
Page #:   1032-1037
Document ID: ISBN 1-555860-070-1
Agency:
Type of Document: Proceedings

Abstract/Citation:
ABSTRACT
  Detection and signaling of occlusion in motion and stereo imagery
has traditionally been based on approaches that involve the
independent, full computation of flow and disparity in the
respective image-pairs. These approaches tend to present a circular
argument to the problem since prior knowledge of occlusion
boundaries is essential for the prevention of smoothing across them
in the computation of optic flow or disparity. In this paper, we
use the information in both the stereo and motion sequences at two
time instances to propose an early indicator of the presence of
occlusion prior to the full computation of flow and disparity.
  Results are demonstrated on real stereo-motion imagery.

File ID: 00051

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Ulupinar, Fatih
Title: Constraints for Interpretation of Perspective Images

-Media: Image Understanding Workshop

Abstract/Citation:
ABSTRACT
    Problem of surface orientation recovery from line drawing in a single image, obtained under perspective projections is studied. Two constraints, shared boundary constraint and the orthogonality constraint previously used in orthogrphic projection are extended to perspective projection. New constraints are derived from observations of parallelism and a new kind of symmetry that we define, called the convergent symmetry. Convergent symmetry in the type of symmetry we get when we project a symmetric object under perspective projection. Unlike skew symmetry, convergent symmetry provides sufficient constraints to recover unique surface orientations. The set of techniques given should allow extension of all previous orthographic analysis and provides new tools for additions, more constrained analysis. An example illustrating the use of our techniques is provided. Finally, extension of the constraints for some class of curved surfaces is dicussed.

File ID: 00052

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:   Perception
Key Word 4:

Author: Weinshall, Daphna
Title: Qualitative Shape from Stereo

_Media: Image Understanding Workshop

Abstract/Citation:
ABSTRACT
   Vision is sometimes described as a problem of inverse optics,
which makes its solution mathematically cumbersome and unstable.
 Human vision does not seem to involve the computation of the
inverse mapping of the projection of 3D world onto a 2D retina but
something more qualatative. This work concentrates on qualitative
shape information that can be obtained from stereo disparities with
little computation. Local surface patches are classified as convex,
concave, hyperbolic or parabolic, using a simple function of image
disparities. The axes of minimum, maximum, and zero curatures are
also obtained. The algorithm works well with synthetic images and
exact disparities. It is used to compute axes of zero curvature on
a real image.

File ID: 00053

Subject: Human Factors
Key Word 2:   Display Technology
Key Word 3:
Key Word 4:

Author: Blanchard, Chuck
Title: Reality Built for Two: A Virtual Reality Tool

Media: VPL Reserarch Inc.

Publication Date:
Publisher, City:
Page #:
Document ID:
Agency:
Type of Document:

Abstract/Citation:
ABSTRACT
   Reserachers have been working with head mounted displays and
virtual reality (VR) since 1965 when Ivan Sutherland published his
first paper on the subject.1 This work has centered on a single
user within virtual space. The literature has covered applications
such as telerobotics, virtual control panels, architectural
simulation and scientific visualization.2,3,4
   Several factors now militate toward broadening the range of VR
application: developments in hardware and software, as well as
growing readiness in many fields to incorporate VR. User interface
constraints of VR systems need to change in response to this
changing user profile. Design of environments and behaviors in
virtual worlds should be simple and accessible to experts in many
fields. With the advent of multi-user systems, communications will
become a major application of virtual reality.
   VPL has recently developed a system that allows more than one user
to share a virtual space. The forms and behaviors of virtual worlds
are specified graphically, so that non-programmers can design them.
 The system, called Reality Built for Two, will be demonstrated on
Interactive 3D graphics.

File ID: 00054

Subject: Human Factors
Key Word 2:   Manipulator
Key Word 3:
Key Word 4:

Author: Thompson, David E.
Title: A Hand Biomechanics Workstation

Media: Computer Graphics

Publication Date: 1-Aug-88
Publisher, City:
Page #: 335-343
Document ID: Vol 22, Number 4
Agency:
Type of Document: Magazine

Abstract/Citation:
ABSTRACT
   Interactive graphics for hand surgery was used to apply
mathematical modeling and describe and knematics of the hand and its
resultant effect on hand function. Dynamic high resolution displays
and three-dimensional images were tailored for use with a specific
patients' hand and a new and powerful design and analysis tool
produced. Methods were developed to portray kinematic information
such as muscle excursion and effective moment arm and extended to
yield dynamic information such as torque and work. This prototype
workstation has been developed in concert with leading orthopedic
surgeons and therapists.

File ID: 00055

Subject: Human Factors
Key Word 2:   Virtual Environment
Key Word 3:
Key Word 4:

Author: Lanier, Jaron, Margret Minsky, Scott Fisher
Title: Virtual Environments and Interactivity: Windows to
         the Future
Media: Siggraph '89 Panel Proceedings

Publication Date: 1-Jan-89
Publisher, City:
Page #:   7-18
Document ID:
Agency:
Type of Document: Slide Show

Abstract/Citation:
Slide show presentation by Jaron Lanier, VPL Research
                          Margret Minsky, University of North
                            Carolina
                          Scott Fisher, NASA Ames Research Center
                          Allison Druin, Tell Tale Technologies

File ID: 00056

Subject: Human Factors
Key Word 2:   Telerobotics
Key Word 3:
Key Word 4:

Author: Anderson, David Elliot
Title: Telerobotics as an EVA Tool

Media: Space Station and Advanced EVA Technologies

Publication Date: 1-Jul-90
Publisher, City: Society of Automotive Engineers, Warrendal
Page #:   43-49
Document ID: SP-830 (book)   901397 (paper ID)
Agency: SAE
Type of Document: book

Abstract/Citation:
ABSTRACT
   Extravehicular activity (EVA) will be required in future space
missions for on-orbit assembley maintenance, and servicing of space
vehicles.   Current EVA activities rely extensively on the use of EVA
crew members for the assembly, maintenance and repair of space
vehicles, satellites and structrues, consequently exposing them to a
variety of hazards, ranging from radiation and impacting debris to
physiological dangers, such as bends and air embolism. Relieving
the crew members of dangerous or time-consuming EVA tasks by
allowing routine or hazardous EVA operations to be conducted by a
telerobotic device would be a significant advance in EVA technology.
 Properly incorporated telerobotic devices will shorten operational
performance schedules, increase crewmember safety, and reduce
staffing requirements, thereby increasing the performance of future
space systems. This paper discusses the possible uses of
telerobotic devices in EVA, drawing from Propellant Tank Farm
neutral buoyancy testing performed under McDonnel Douglas
independent research and development (IRAD). Recommendation are
made for using telerobots such as the Flight Telerobotic Servicer,
the Special Purpose Dextrous Manipulator, the Remote Manipulator
System, and the Japanese Experiment Module arms in the evolution of
Space Station Freedom.

File ID: 00057

Subject: Human Factors
Key Word 2:   Gloves
Key Word 3:
Key Word 4:

Author: Spampinato, Phil
Title: Advanced Technology Application in the Production of
        Sapcesuit Gloves
-Media: Space Station and Advanced EVA Technologies

Publication Date: 1-Jul-90
Publisher, City: Society of Automotive Engineers, Warrendal
Page #:   35-42
Document ID: SP-830 (book)   901322 (paper ID)
Agency: SAE
Type of Document: book

Abstract/Citation:
ABSTRACT
   ILC Dover successfull designed and developed an advanced high
pressure (8.3 psia) Spaceauit Glove for use on the space station.
 As an aide to fabricaion of this glove, a feasibility study has
been performed to use laser or photo optical, non-contact scanning,
CAD and CAM technologies.
   The current process for fabrication of spacesuit gloves starts by
taking hand cast of a crewman's hands in one or more positions. The
castings are subsequently measured by hand in critical areas, and a
manual system of defining the glove bladder and glove restraint
patterns follows.
   The proposed process will involve collecting dimensional data on
hand using laser or photo optical scanning techniques. Key
dimensions will be identified on a CAD system. Algorithms
pre-programmed in the CAD system along with some CAD modeling will
be used to manipulate the scanned data to define the glove bladder
and glove restraint. Separate CAD subroutines will subsequently
define the glove bladder dip form and glove restraint patterns. The
dip form data will then be directed to a Stereolithography machine to
produce a dip form. As a result of this study, a method of
providing a more accurate fit, in less time, will be defined. A
goal of the study is to cut in half the current time required to
deliver a pair of customized flight gloves.

File ID: 00058

Subject: Human Factors
Key Word 2:   Machine Interaction
Key Word 3:
Key Word 4:

Author: Smith, Jeff
Title: Applying the Interaction Design Approach to Medical
        Devices
Media: Medical Design and Material

Publication Date: 1-Apr-91
Publisher, City: Aster Publishing Corp., Eugene, Oregon
Page #:   50-55
Document ID: vol 1, number 4
Agency:
Type of Document: Magazine

Abstract/Citation:
INTRODUCTION
   Stated simply, the goal of interaction design is to create
friendlier and more efficient interfaces between humans and machines
or processes. This dicipline can be extremenly useful in the design
of medical devices and diagnostics often incorporate highly
complicated and specialized technology that must be made easily
accessible and sage. Interaction design seeks to unify physical anc
visual processes and intuiutive and cognitively simple
charactersitics to simplify or facilitate a particular process or
action. For medical equipment manufacturers, interaction design
can translate into speed (in both learning time and throughput),
accuracy, and accessibility to state-of-the-art medical tachnology.

File ID: 00059

Subject: Human Factors
Key Word 2:   Wide-Angle Display
Key Word 3:
Key Word 4:

Author: Fetter, William A.
Title: Wide-Angle Display Developments by Computer Graphics

Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   47-1 -- 47-5
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
SUMMARY
   Computer graphics can now expand its new subset, wide-angle
projection, to be as significant a generic capability as computer
graphics itself. My purpose is to present you with some prior work
in computer graphics leading to an attractive further subset of
wide-angle projection, called hemispheric projection, to be a major
communication media. Hemisphereic film systems have long been
present and such computer graphics systems are in use in simulators.
 This is the leading edge of capabilities which should ultimately be
as ubiquitous as CRTs. The credentials I have for making these
assertions are not from degrees in science or only from my degree in
graphic design, but in a history of computer graphics innovations,
laying groundwork by demonstration. I believe it is timely to look
at several development strategies, since hemisperic projection is
now at a point comparable to the early stages of computer graphics,
requiring similar patterns of development again.

File ID: 00060

Subject: Human Factors
Key Word 2:   Telemanipulation
Key Word 3:
Key Word 4:

Author: Hannaford, Blake
Title: Displays for Telemanipulation

Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #: 32-1 -- 32-17
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
SUMMARY
  Visual displays drive the human operator's highest bandwidth
sensorry input channel. Thus, no telemanipulation system is
adequate which does not make extensive use of visial displays.
  Although an important use of visual displays is the presentation of
a televised image of the work scene, this paper will concentrate on
visual displays for presentation of nonvisual information (forces
and torques) for simulation and planning, and for management and
control of the large numbers of subsystems which make up a modern
telemanipulation system.

File ID: 00061

Subject: Human Factors
Key Word 2:   Manipulation
Key Word 3:
Key Word 4:

Author: Kim, Won S.
Title: Visual Enhancements in Pick-and Place Tasks: Human
          Operators Controlling a Simulated Cylindrical
-Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   30-1 -- 30-25
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:

ABSTRACT
  A visual display system serves as an important human/machine
interface for efficient teleoperations.   However, careful
consideration is necessary to display three-dimensional information
on a two-dimensional screen effectively.   A teleoperation simulator
is constructed with a vector display system, joysticks, and a
simulated cylindrical manipulator in order to evaluate various
display conditions quantitatively.   Pick-and-place tasks are
performed, and mean completion times are used as a performance
measure.   Two experiments are performed.   First, effects of variation
of perspective parameters on a human operator's pick-and-place
performance with monoscopic perspective dispaly are investigated.
  Then, visual enhancements of monoscopic perspective display by
adding a grid and reference lines are investigated and compared with
visual enhancements of stereoscopic display.   THe results indicate
that stereoscopic display does generally permit superior
pick-and-place performance, while monoscopic display can allow
equivalent performance when it is defined with appropriate
perspective parameter values and provided with adequate viaual
enhancements.   Mean-completion-time results of pick-and-place
experiments for various display conditions shown in this paper are
observed to be quite similar to normalized root-mean-sqare error
results of manual tracking experiments reported previously.

File ID: 00062

Subject: Human Factors
Key Word 2:   Telemanipulation
Key Word 3:
Key Word 4:

Author: Held, Richard
Title: Telepresence, Time Delay, and Adaptation

-Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   28-1 -- 28-15
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
INTRODUCTION
  Displays, which are the subject of this conference, are now being
used extensively throughout our society. More and more of our time
is spent watching television, movies, computer screen, etc.
  Furthermore, in an increasing number of cases, the observer
interacts with the display and plays the role of operator as well as
observer. To a large extent, our normal behavior in our normal
environment can also be thought of in these same terms. Taking
liberties with Shakespeare, we might say the " all the world's a
display and all the individuals in it are operators in and on the
display."
  Within this general context of interactive cisplay systems, we
being our discussion with a conceptual overview of a particular
class of such systems, namely teleoperator systems. We then
consider the notion of telepresence and the factors that limit
telepresence, including decorrelation between the (1)motor output
of the teleoperator as sensed directly via feedback from the slave
robot, i.e., via a visual display of the motor actions of the slave
robot. Finally, we focus on the deleterious effect of time delay
(a particular source of decorrelation) on sensory-motor adaptation
(an important phenomenon related to telepresence).

File ID: 00063

Subject: Human Factors
Key Word 2:   Telemanipulation
Key Word 3:
Key Word 4:

Author: McKinnon, G.M.
Title: Multi-Axis Control of Telemanipulators

-Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   27-1 -- 27-11
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
ABSTRACT
   This paper describes the development of multi-axis hand
controllers for use in telemanipulator systems. Experience in the
control of the SRMS arm is reviewed ogether with subsequent tests
involving a number of simulators and configurations, including use
as a side-arm flight control for helicopters. The factors affecting
operator acceptability are reviewed.

INTRODUCTION
   The sucess of in-orbit operations depends on the use of autonomous
and semiautomonous devices to perform construction, maintenance and
operational tasks. While there are merits to both full autonomous
and man-in-the-loop (or teleoperated) systems, as well as for pure
extravehicular activity (EVA), it is clear that for many tasks, at
least in early stages of development, teleoperated systems will be
required.
   This paper reviews some experience gained in the design of the
human machine interface for teleoperated systems in space. A
number of alternative approaches have been proposed and evaluated
over the corse of the work described, and some basic design
principles have evolved which may appear mundane or obvious after
the fact, but which nevertheless are critical and often ignored.
   One key design objective in the implementation of human-machine
interfaces for space is that of standardization. Astronauts should
naturally and comfortably interpret their input motions in terms of
motions of the manipulator or task. This "transparency" is achieved
by careful design to ensure that task coordinates and views are
always presented in a clear, unambiguous and logical way, and by
ensuring that standardized input devices are use in standardized
modes. If conventions are established and systematic modes of
control are respected, training time is reduced and effectiveness
and performance are improved. The end objective in the design of
displays and controls for telemanipulators is to establish a "remote
presence" for the operator.

File ID: 00064

Subject: Human Factors
Key Word 2:
Key Word 3:
Key Word 4:

Author: Hart, Sandra G.
Title: Helmet-Mounted Pilot Night Vision Systems: Human
         Factors Issues
Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   13-1 -- 13-21
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
ABSTRACT
  Helmet-mounted displays of infrared imagery (forward-looking
infrared (FLIR)) allow helicopter pilots to perform low-level
missions at night and in low visibility. However, pilots experience
high visual and cognitive workload during these missions, and their
performance capabilities may be reduced. Human factors problems
inherent in existing systems stem for three primary sources: (1) the
nature of thermal imagery, (2) the characteristics of specific FLIR
systems, and (3) the difficulty of using FLIR system for flying
and/or visually acquiring and tracking objects in the environment.
  The pilot night vision system (PNVS) in the Apache AH-64 provides a
monochrome 30 deg by 40 deg helmet-mounted display of infrared
images. Thermal imagery is inferior to television imagery in both
resolution and contrast ratio. Gray shades represent temperatures
differences rather that brightness variabilty, and images undergo
significant changes over time. The limited field of view,
displacement of the sensor from the pilot's eye position, and
monocular presentation of a bright FLIR image (while the other eye
remains dark-adapted) are all potential sources of disorientation,
limitations in depth and distance estimation, sensations of apparent
motion, and difficulties in target and obstacle detection.
  Insufficient information about human perceptual and performance
limitation restrains the ability of human factors specialists to
provide significantly improved specification, training programs, or
alternative designs. Additional research is required to determine
the most critical problem areas and to propose solutions that
consider the human as well as the development of technology.

File ID: 00065

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:
Key Word 4:

Author: Foley, John F.
Title: Stereoscopic Distance Perception

-Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   3-1 -- 3-5
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
INTRODUCTION
  Most of this article is concerned with limited cue, open-loop
tasks in which a human observer indicates distance or relations
among distances. By open-loop tasks I mean tasks in which the
observer gets no feedback as to the accuracy of responses. At the
end of the article, I will consider what happens when cues are added
and when the loop is closed, and what the implications of this
research are for the effectiveness of visual displays.
  Errors in visual distance tasks do not necessarily mean that the
percept is in error. The error could arise in transformations that
intervene between the percept and the response. I will argue,
however, the the percept is in error. I will argue further that
there exist post-perceptual transformations that may contribute to
the error or be modified by feedback to correct for the error.

File ID: 00066

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:
Key Word 4:

Author: Schor, Clifton
Title: Spatial Constraints of Stereopsis in Video Displays

-Media: Spatial Display and Spatial Instruments Proceedings

Publication Date: 31-Aug-87
Publisher, City: NASA Ames, Moffett Field, CA
Page #:   2-1 -- 2-14
Document ID: NASA CP-10032
Agency: NASA
Type of Document: Paper

Abstract/Citation:
Recent development in video technology, such as the liquid-crystal
displays and shutters, have made it feasible to incorporate
stereoscopic depth into the three-dimensional representations on
two-dimensional displays. However, depth has already been vividly
portrayed in video displays without steropsis using the classical
artists depth cues described by Helmholtz (1866) and the dymanic
depth cues described in detail by Ittleson (1952). Successful
static depth cues include overlap, size, linear perspective, texture
gradients, and shading. Effective dynamic cues include looming
(Regan and Beverly, 1979) and motion parallax (Rogers and Graham,
1982).

File ID: 00067

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:
Key Word 4:

Author: Beaton, Robert
Title: Displaying Information in Depth

Media: SID Digest 90

Publication Date: 1-Jan-90
Publisher, City:
Page #:   355-358
Document ID: ISSN0097-0966X/90/0000-35
Agency: Society for Information Display
Type of Document: Magazine

Abstract/Citation:
ABSTRACT
   This paper examines the utility of monocular and binocular depth
cues in electronic 3-D display systems.   In several experiments,
the effects of depth cues were assesed in various task contexts,
including relative depth judgements, visual search, cursor
positioning, and subjective image quiality judgments.   The results
support the notion that binocular disparity cues enhance human
performance on task involving complex information formats.

File ID: 00068

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:   Cognition
Key Word 4:

Author: editors - Boff, Kenneth R., Kaufman, Lloyd
Title: Handbook of Perception and Human Performance   vol II

-Media: collection of papers

Publication Date: 1-Jan-86
Publisher, City: John Wiley and Sons, New York, NY
Page #:
Document ID: ISBN 0-471-82957-9 (vol II)
Agency:
Type of Document: Book

Abstract/Citation:
This is a collection of studies on the Cognitive Processes and
Performance. A table of Conntents is located on file.

File ID: 00069

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Chase, William G.
Title: Chapter 28, Visual Information Processing

_Media: Handbook of Perception and Human Performance   vol II

Abstract/Citation:
   p.28-2
   The central concern of the information-processing approach is two
fold: (1) to specify the mental structures underlying the
representation of perceptual knowledge, and (2) to specify the
mental processes that operate on these structures.

   One of the goals of this chapter is to illustrate the
information-processing methodology in selected areas where this
approach shows some promise.
   Feature theories postulate that the visual system analyzes and
represents sensory information in abstract, primitive informational
units called features or attributes. Feature theories represent a
more realistic psychological appproach to pattern recognition
because there is so much evidence that detection of distinctive
features is an integral part of the recognition process. A
distinctive feature can be used to make a critical distinction
between two patterns or classes of patterns. Most of the
theoretical work on feature theories has been on verbal materials,
such as speech(Chomsky & Halle, 1968), Morse code (Selfridge &
Neisser, 1960), or letter of the alphabet (Gibson, 1969).
   Thus there is overwhelming evidence that feature extraction is a
necesssary part of the process of pattern recognition.

   p.28-4
   Informational content is simply a count of the number of
structural symbols contained in the symbolic representation,
excluding metrical quantities that determine the scale but have no
beáring on the shape.

   p.28-5
   The results of these two experiments are good evidence that
Leeuwenburg's (1968) coding system predicts the Gestalt law of
pragnanz, perceptually preferred interpretations are the most
efficient interpretations.

9/20/91   Page 73

File ID: 00069

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Chase, William G.
Title: Chapter 28, Visual Information Processing

-Media: Handbook of Perception and Human Performance   vol II

Publication Date: 1-Jan-86
Publisher, City: John Wiley and Sons, New York, NY
Page #:   28-1 -- 28-68
Document ID: ISBN 0-471-82957-9 (vol II)
Agency:
Type of Document: chapter

Abstract/Citation:
   p.28-13
   A frame is a structure for organizing familiar situations;
universal features of the situation are fixed in the framework of the
structure, and new or variable features are assigned "slots".
   Minsky (1975) originally developed frame theory to solve some very
fundamental spatial problems, such as keeping track of the different
perspective appearances of an object, representing the insides of
rooms and moving around with in a house.

   p.28-14
   A new frame is created, but the knowledge of the previous view is
saved, and the two frome are linked together into a frame systems by
common variable bindings. Now, if the perspective moves back to the
original perspective, the first scene can be reconstructed without
any new perceptual computation (fig 28.17c). As the cube is viewed
from different perspectives, a more elaborate frome system is built
to represent the object from different perspectives to represent
operations on the object (e.g. MOVE-RIGHT) and the perceptual
consequences (fig 28.17d). Once such a frame is learned and stored
in memory, it can be used to perceive cubes and cubelike objects in
new situation and to guide perceptual expectations.

   p.28-15
   These results are good evidence that people represent a class of
patterns with a prototype, and new patterns are classified in part
according to how similar they are to the prototype.

   p. 28-19
   Table 28.2
   On each trial of a speeded verification task, subjects were
presented a category name at one of three levels of generality:
superordinate, basic, or subordinate. After a 500-msec blank
interval, a picture of an object was displayed and subjects were to
decide as quickly as possible if the pictured object belonged in
the presented category. Mean RT's are based on 15 subjects in each

File ID: 00069

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Chase, William G.
Title: Chapter 28, Visual Information Processing

-Media: Handbook of Perception and Human Performance   vol II

Publication Date: 1-Jan-86
Publisher, City: John Wiley and Sons, New York, NY
Page #:   28-1 -- 28-68
Document ID: ISBN 0-471-82957-9 (vol II)
Agency:
Type of Document: chapter

Abstract/Citation:
of the three category conditions receiving 18 "true" trials and 18
"false" trials. Results show that subjects performed basic-level
judgements more quickly than either superordinate- or
subordinate-level category judgements both for "true" and "false"
trials. The implication is that objects are first perceived at a
basic categorical level and that additional processing is required
to identify them with their appropriate superordinate or subordinate
categories.

File ID: 00070

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: eds. Boff, Kenneth R., Lloyd Kauffman
Title: Handbook of Perception and Human Performance   vol I

-Media: book

Publication Date: 1-Jan-86
Publisher, City: John Wiley and Sons, New York
Page #:
Document ID: ISBN 0-471-88544-4 (vol. I)
Agency:
Type of Document: colectn of research

Abstract/Citation:
Table of contents for the book is located on file.

File ID: 00071

Subject: Human Factors
Key Word 2:   Stereoscopic
Key Word 3:   Perception
Key Word 4:

Author: Arditi, Aries
Title: chapter 23, Binocular Vision

Media: Handbook of Perception and Human Performance vol. I

Publication Date: 1-Jan-86
Publisher, City: John Wiley and Sons, New York
Page #:   23-1 -- 23-36
Document ID: ISBN 0-471-88544-4 (vol. I)
Agency:
Type of Document: Chapter from book

Abstract/Citation:
This chapter discusses the phenomena of stereoscopy as well as
techniques and methods of achieving a stereoscopic display.

File ID: 00072

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Ellis, Stephen R.
Title: Visions of Visualization Aids: Design Philosophy
         and Observations
Media: Proceedings of the SPIE-The International Society
         for Optical Engineering
Publication Date: 15-Jan-89
Publisher, City: SPIE, Los Angeles, CA
Page #:   220-227
Document ID: SPIE vol. 1083
Agency:
Type of Document: paper

Abstract/Citation:
ABSTRACT
   Aids for the visualization of high dimensional scientific or
other data must be designed. Simply casting multidimensional data
into a 2 or 3D spatial metaphor does not guarantee that the
presentation, in contrast to slityz, high-tech, computer-graphics
imagery, is generally based on pre-existing theoretical beliefs
concerning the underlying phenomena. These beliefs guide selection
and formatting of the plotted variables.
   Visualization tools are useful for understnading naturally 3D
databases such as those used by pilots or astronauts. Two examples
of such aids for spatial maneuvering illustrate that informative
geometric not be adequate to provide the necessary insight into the
underlying processes.

File ID: 00073

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:   Field of View
Key Word 4:

Author: Fischer, S. S.
Title: Virtual Interface Environment Workstations

-Media: Proceedings of the Human Factors Society - 32nd
               Annual Meeting 1988
Publication Date: 1-Jan-88
Publisher, City:
Page #:   91-95
Document ID:
Agency:
Type of Document: paper

Abstract/Citation:
ABSTRACT
   A head-mounted, wide-angle, stereoscopic display system controlled
by operator position, voice and gesture has been developed at NASA's
Ames Research Center for use as a multipurpose interface
environment.   This Virtual Interface Environment Workstation (VIEW)
system provides s multisensory, interactive display environment and
can viscerally interact with its components.   Primary applications
of the systems human factors research.   System configuration,
research scenarios, and research directions are described.

   VIEW Visual Display Technology
   The VIEW Display subsystem is....image display elements, optics,
and electronics which together probide a wide angle, stereoscopic
image environment that closelt matches human binocular visual
capabilities.   When combined with high-resolution, magnetic 6 degree
of freedom head and limb position tracking technology, the
disp;layed iimagery appears to completely surround the user in
3-space and provides interactive viewing and manipulation
capabilities.   .....

   OPTICS
   .....These optics provide a 120 degree horizontal and vertical
field of view for each eye and up to a 90 degree binocular field
overlap.   Total instantaneous visual field of view is approximately
120 degrees.   The 2.75" diameter of each optical element requires a
minimum 4" diagonaly display size to completely fill the available
field of view.   With a 640x220 pixel resolution, each pixel in the
diagonal array (i.e. each distinguishable vertical line) subtens .38
degrees (22.5 minutes) of horizontal visual field.   ...imagery that
requires a barrel distortion compensation in the image generation or
image capture technology for correct scene representation.

   MOUNTING CONFIGURATION
   .....The entire unit is counterweighted to transfer the weight of

File ID: 00073

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:   Field of View
Key Word 4:

Author: Fischer, S. S.
Title: Virtual Interface Environment Workstations

_Media: Proceedings of the Human Factors Society - 32nd
            Annual Meeting 1988
Publication Date: 1-Jan-88
Publisher, City:
Page #:   91-95
Document ID:
Agency:
Type of Document: paper

Abstract/Citation:
the display and optics unit to a point directly over the operator's
spine.

   RESEARCH SCENARIOS
          Telerobotics
   ....Switching to telepresence control mode, the operator's
wide-angle, stereoscopic display is directly linked the the
telerobot 3D camera system for precose viewpoint control.

File ID: 00074

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:   Field of View
Key Word 4:

Author: Fisher, Scott S.
Title: Living in a Virtual World

_ Media: Byte Magazine

Publication Date: 1-Jul-90
Publisher, City: Byte
Page #: 215-221
Document ID:
Agency:
Type of Document: article

Abstract/Citation:
   p.216
   ....The scientists at Ames have developed the Virtual Interface
Environment Workstation (VIEW), a wide-angle, head-mounted ,
stereoscopic display system that the operator's voice, posistion,
and gestures control (see photo 1). This system enables you to
explore all 360 degrees of a virtual environment and interact with
it in various ways.

   p.218
   ...The currnet size of each display if 3.9 inches (on the
diagonal) with a 4-3 aspect ratio....640-220 pixels with
approximately 16 levels of gray scale.... viewed through a pair of
wide-angle magnifying lenses mounted about 5 milimeter from the
screens. ....The feeling of being in a virtual workld requires that
the field of vision closley resembles that of human binocular
vision....The displays must competely fill your field of vision to
give you a true sense of being present in the virtual environment
(120 degrees both horizontally and vertically, with up to 90 degrees
overlap in the binocular fields.) The headset also contains a
tracking device that detects where within the environment you are
looking. Currently, NASA Ames is using an eletrmagnetic device for
this purpose, one the can determine where the head is withis a
magnetic field. ...With this information the system can refresh the
image shown to one that matches the position of your head. New
images are drawn so quickly that is feels as if you're really there.
   p.220
   .....Eye tracking will reveal the point at which your eyes
converge - that is, where they are focused. Thus you could pull
donw menus or move a cursor simply by looking at certain point. In
addition for depth-of-field information in crease when eye tracing
is used.

File ID: 00075

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Morris, Ailene
Title: The Time Required for U.S. Navy Fighter Pilots to
        Shift Gaze and Identify Near and Far Targets
_Media: Aviation, Space, and Evironmental Medicine

Publication Date: 1-Nov-89
Publisher, City: Aerospace Medical Association, Washinton D
Page #: 1085-1089
Document ID: 60:1085-9
Agency:
Type of Document: paper

Abstract/Citation:
RESULTS
   Subject age was a significant determinant of the time required to
shift gaze. As age increased, speed slowed.   ...The slope of the
regression showed that the mean Near-to-Far speed slowed 0.0196 s
per year.
   .....The slope of the regression show the the mean Far-to-Near speed
slowed 0.0615 s per year, which is more thatn 3 times greater the
the slowing seen with the Near-to-Far data.

   p.1087
   ...Far-to-Near speeds measured with the 2.0mva targets. With this
size stimulus, Far-to-Near speed slowed with age at a rate of about
0.009 s per year ..... The mean Far-to-Near time measured with the
2.0-mva target was 0.2869 s while with the 1.0 mva the mean time was
0.6728 s, a statistically significant difference......With the 1.0
mva stimuli, Far-to-Near standard devieations increased 0.0125 s per
year. The increase is standard deviation with age, measured with
the 2.0 mva target was 0.0035 s per year. Near-to-Far standard
deviations did not increase with age.

File ID: 00076

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Abstract/Citation:
INTRODUCTION
  The choice of the viewpoint and field of view for such displays
highly depends on the function to be performed.   Thus, the view
point can be located either in the cockpit, producing a
through-the-windshield scene, e.g. as in the "tunnel" display or
[1] used for path-following in the terminal area, or can be located
slightly above and behind the aircraft, producing a birds-eye view,
as in the "pathway-in-the-sky" display or [2], it can be far above
and behind the aircraft, as in the airborrne traffic display of
[3]...... Their interpretation requires the mental recontruction of a
three dimensional situation from the two-dimensional perceived
image.   The errors involved in this process largely depend on the
correct choice of the relevant viewing parameters, such as
view-point, field of view, level of detail, etc.   In this paper the
development of the spatial orientation model has been motiviated by
the need to: 1) analyze and understand the process involved in
reconstructing a three dimensional situation form a two-dimensional
image; 2)investigate how basic assumption about the characteristics
of the objects are used in the process; an 3) obtain a better
understanding of how visual parameters such as viewpoint location
and field of view affect the estimation errors and accuracies....
  Familiarities with the shape of an object provides an essential
cue for perceiving its spatial orientation.   If the observer has
some apriori knoweledge of the perceived objects, e.g. size, shape
or characteristics like parallelism or perpendicularity of lines or
planes, he is able to utilize this knowledge to estimate the spatial
position and angular orientation of these objects....Other examples
of the importantance of familiarity cues are found in the well known
demonstrations of A. Ames [4], which show how commonplace
assumptions of object shape can lead to substantial perceptual
errors in judging size and orientation of appropriately distorted
objects.   The usefulness of these familiarity cues for perceiving
the orientation of simple objects, such as a cube, has been
previously investigated by Attneave and Frost [5],[6].   They showed
how assumptions about the object's shape can be used to allow an

File ID: 00076

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Grunwald, Arthur J.
Title: A Mathematical Model for Spatial Orientation form
         Pictorial Perspective Displays
Media: IEEE Transactions on Systems, Man, and Cybernetics

Abstract/Citation:
observer to reconstrut the object's orientation from the geometry of
its two-dimensional projection. ....In contrast to McGreevy's and
Farber and Rosinski's models, the computation process presented in
this paper utilized the known function relationship between the
coordinates of the object. All the points of the object are treated
simultaneously and rather than being distorted in the virtual space,
the object will be mismatched with the lines-of-sight instead.
   ...Systematic errors in the perception of the lines-of-sight
include errors in the relative scaling of the perceived image.
  These errors are modelled by expanding or shrinking the size of the
image about the viewing axis xe, as done with the zoom operation of
a camera lens.
  The complex process of reconstruction the three dimensional
situation from the perceived lines-of-sight and on the basis of
the a priori knowledge of the object characteristics involves
several parts: 1) determination of unknown parameters such as
viewpoint location and orientation of the azimuth plane with respect
to the ground plane; 2) estimation of the spacing between the lines
in the groung grid; and 3) estimation of the dimensions of the
trapezoid. The algorithm of the reconstruction process is outlined
hereafter.
  ANALYTICAL STUDIES
  The purpose of the analytical study was to investigate the effect
of view point location, object shape, LOS and shape noise
magnitudes, and image scaling errors on the model esitmates.

  CONCLUSIONS
  The analytical model is a useful tool for analyzing and developing
pictorial perspective displays presenting clearly outlined objects
composed of a limited number of lines or polygons....Familiar
characteristics such as parallelism or perpendicualrity of lines or
planes provide essential cues for reconstructing a spatial layout
from perspective projections. The spatial reconstruction process is
subject to the observer's basic assumptions about the object
characteristics. Incorrect assumptions may result in substantial

File ID: 00076

Subject: Human Factors
Key Word 2:   Perception
Key Word 3:
Key Word 4:

Author: Grunwald, Arthur J.
Title: A Mathematical Model for Spatial Orientation form
         Pictorial Perspective Displays
Media: IEEE Transactions on Systems, Man, and Cybernetics

Abstract/Citation:
estimation errors.   .....Therefore, inorder to minimize
systematic errors, the center of projection of the display should be
placed somewhat in front of the eye position, i.e. the image should
present a wider field of view tha requires by the correct viewing
geometry. This comfirms the results obtained earlier by McGreevy
and Ellis {8],[9]. See Roscoe et al. [11] for a report of a similar
but opposite wide-angle perceptual bias......Judgement errors are
the smallest in the grid axis directions. An orthogonal ground grid
with equidistant spacing is expected to be a useful element in
perspective displays. For objects not located in the ground plane,
it is useful to display the line segments connecting the objects to
the ground plane.
   The presence of motion is expected to improve the estimates
considerably and its incorporation in the model is a subject of
further study.

File ID: 00077

Subject: Human Factors
Key Word 2:   3D Acoustics
Key Word 3:
Key Word 4:

Author: Strybel, Thomas
Title: Perception of Real and Simulated Motion in the
         Auditory Modality
Media: Proceeding of the Human Factors Society 32nd Annual
         Meeting
Publication Date: 1-Jan-88
Publisher, City:
Page #:   76-80
Document ID:
Agency: Human Factors Society
Type of Document: paper

Abstract/Citation:
ABSTRACT
   Future head-coupled display systems will include auditory spatial
information in order to direct the pilot's attention to critical
events in the environment. It is anticipated that such a system
will provide dynamic as well as static auditory location
information. This report reviews current research in the area of
auditory motion perception, particularly as it applies to the
development of simulated 3-dimensional auditory space.

File ID: 00078

Subject: Human Factors
Key Word 2:   3D Acoustics
Key Word 3:
Key Word 4:

Author: Wenzel, Elizabeth M.
Title: A Virtual Display System for Conveying Three-
         Dimensional Information
-Media: Proceedings of the Human Factors Society 32nd Annual
         Meeting
Publication Date: 1-Jan-88
Publisher, City:
Page #:   86-90
Document ID:
Agency: Human Factors Society
Type of Document: paper

Abstract/Citation:
ABSTRACT
   A three-dimensional auditory display could take advantage of
intrinsic sensory abilites like localization and perceptual
organization by generating dynamic, multidimensional patterns of
acoustic events that convey meaning about objects in the spatial
world. Applications involve any context in which the user's
situational awareness is critical, particularly when visual cues are
limited or absent; e.g., display would generate localized cues in a
flexible and dynamic manner. Whereas this can be readily achieved
with an array of real sound sources or loudspeakers, the NASA-Ames
prototype maximizes flexibility and portability by synthetically
generating three-dimensional sound in realtime for delivery through
headphones. Psychoacoustic research suggests that
perceptually-verdical localization over head-phones is possible if
both the direction -dependant pinna cues and the more well
understood cues of interaural time and intensity are adequately
synthesized. Although the realtime device is not yet complete,
recent studies at the Univeristy of Wisconsin have confirmed the
perceptual adequacy of the basic approach to synthesis.

File ID: 00079

Subject: Human Factors
Key Word 2:   Voice Recognition
Key Word 3:   Head Mounted Display
Key Word 4:

Author: Shepard, C.K.
Title: The Helmet-Mounter Display as a Tool to Increase
        Productivity During Space Station Extravehicular
Media: Proceedings of the Human Factors Society 32nd Annual
        Meeting
Publication Date: 1-Jan-88
Publisher, City:
Page #:   40-43
Document ID:
Agency: Human Factors Society
Type of Document: paper

Abstract/Citation:
ABSTRACT
   While the debate continues about the safety and applicability of
heads-up displays (HUDs) and helmet-mounted displays (HMDs) in the
aeronautical environment (as demonstrated in the July, October, and
November 1987 issues of the Human Factors Society Bulletin), a
voice-controlled HMD is being designed as describes the human
factors issues that suggest the HMD will be a safe and desirable
tool for Space Station extravehicular activity (EVA). Also, it
briefly outlines a Macintosh-based voice-interactive rapid
prototyping system that is being used at the NASA Johnson Space
Center for simulation and evaluation the HMD's ability to enhance
astronaut productivity in the EVA setting.

     ......To enhance productivity, it is important that the crewmember
have his hands free for his work,.....In addition, the hands
frequently are occupied with restraint tasks, i.e. grasping a
handhold for self-restraint, or restraining a tethered tool from
drifting into the work area. Therefore, ideally , a EMU information
system should provide "Hands-free" access to information.......are
few methods but ...voice controlled helmet-mounted display is one
solution that supports the required human factors criteria which will
also meeting other engineering and fiscal considerations.
     ....The visual information required for EVA does not involve
coordination of a computer-generated image on top of the real-world
scene. Thus, problems associated with disorientation and perceptual
errors maybe avoided. During tasks are procedually oriented and are
performed at a comfortable pace, in contrast to the faster paced,
tension-filled environment of the fighter cockpit that demands the
constant vigiliance of the pilot. For these same reasons, optical
minification is not expected to present significant problems.
     ...First, sounds within the EMU are absent of the noises that
commonly cause problems for speech recognizers, such as background
voices and extraneous clamor found at industrial workstation or
breathing noises inherent with masked fighter pilots....Second, an

File ID: 00079

Subject: Human Factors
Key Word 2:   Voice Recognition
Key Word 3:   Head Mounted Display
Key Word 4:

Author: Shepard, C.K.
Title: The Helmet-Mounter Display as a Tool to Increase
         Productivity During Space Station Extravehicular
-Media: Proceedings of the Human Factors Society 32nd Annual
         Meeting
Publication Date: 1-Jan-88
Publisher, City:
Page #:   40-43
Document ID:
Agency: Human Factors Society
Type of Document: paper

Abstract/Citation:
extensive vocabulary of about 50 words will suffice for the
procedurally-oriented tasks of EVA..... Finally, the voice
recognition system need not be speaker-independant since each
EVA-qualified crewmember aboard the Space Station will be assigned
an individual EMU.

File ID: 00080

Subject: Human Factors
Key Word 2:   Eye Movements
Key Word 3:   Eye Tracking
Key Word 4:

Author: McElligott, J.G.
Title: The Use of Synchronous Demodulation for the
        Measurement of Eye Movemnts by Means of an Ocular
-Media: IEEE Transactions on Biomedical Engineering

Publication Date: 1-Jun-79
Publisher, City: IEEE
Page #:   370-374
Document ID: 0018-9294/79/0600-0370$00.75
Agency: IEEE
Type of Document: paper

Abstract/Citation:
ABSTRACT
   This paper descirbes an eye position monitoring system that can be
built using inexpensive components for the demodualtor. It uses a
two-frequency (f and 2f) technique which provides excellent channel
separation with a minimum of adjustment and wide dynamic signal
range. Phasing errors produce no crosstalk effects for a pure
sine-wave signal. It should be noted that this system will also
work as a single-frequency two-phase system, merely by changing the
driver unit.
   A laser reflection technique was used to accurately calibrate the
instrument parameters. A statistical behavioral technique is
described which is routinely used to calibrate the search coil
implanted in the animal.

File ID: 00081

Subject: Human Factors
Key Word 2:   Eye Movements
Key Word 3:   Eye Tracking
Key Word 4:

Author: Kenyon, Robert V.
Title: A Soft Contact Lens Search Coil for Measuring Eye
        Movements
Media: Vision Research vol 25, No. 11

Publication Date: 1-Jan-85
Publisher, City: Pergamon Press Ltd.
Page #: 1629-1633
Document ID: 0042-6989/85 $3.00+0.00
Agency:
Type of Document: paper

Abstract/Citation:
ABSTRACT
   A five turn coil of magnet wire sandwicked between two soft
contact lenses served as a magnetic serch coil to measure horizontal
and vertical eye movements in humans. The lens was adgered to the
eye for 35 min by periodically misting the eye with distilles water;
during this time the records of eye position showed that the lens
remained firmly attached to the eye. Using this sensor, no topical
anesthetic need be applied and the lens can be worn for an extended
period of time without increasing intraocular pressure.

File ID: 00082

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Hacisalihzade, Selim S.
Title: Automatic Analysis of Eye Movements with a PC

Media: IEEE Engineering in Medicine & Biology Society 11th
          Annual International Conf.
Publication Date: 1-Jun-89
Publisher, City: IEEE
Page #:
Document ID: CH2770-6/89/0000-0652 $01.00
Agency: IEEE
Type of Document: paper

Abstract/Citation:
ABSTRACT
   A program package for the acquisiton, analysis and plotting of eye
movements is introduced. The eye movements are measured by means
of an infrared reflectometry technique and recorded with a digital
computer. In order to perform a reasonable quantitative analysis,
calibration is necessary. The package and the signal processing
techniques described in this paper can be used by researchers in a
large variety of fields.

File ID: 00083

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Remmel, Ronald S.
Title: An Inexpensive Eye Movement Monitor Using the
         Scleral Search Coil Technique
Media:

Publication Date:
Publisher, City:
Page #:
Document ID:
Agency:
Type of Document: paper to be submitted

Abstract/Citation:
ABSTRACT
  For the economical recording of eye movements using the scleral
search-coil method, we developed a circuit using common components
and costing only $300. The Helmhotz field coils are 51 cm in
diameter and separated by 51 cm. The horizontal and vertical field
coils are driven at 50 and 75 KHz, respectively, by phase-locked
spare waves; no tuning is required. After amplification the
eye-coil signal is phase detected at 50 and at 75 KHz to
produce the horizontal and vertical eye-position signals,
respectively. For a 2 cm diameter eye coil of one turn, the noise
is 1.0 min of angle rms.

File ID: 00084

Subject: Human Factors
Key Word 2:   Eye Movement
Key Word 3:   Eye Tracking
Key Word 4:

Author: Reulen, J.P.H.
Title: The Measurment of Eye Movement Using Double Magnetic
         Induction
Media: IEEE Transactions on Biomedical Engineering vol. BME-
         29 no. 11
Publication Date: 1-Nov-82
Publisher, City: IEEE
Page #: 740-744
Document ID: 0018-9294.82.1100-0740$00.75
Agency: IEEE
Type of Document: article

Abstract/Citation:
ABSTRACT
   This paper describes a new method for the accurate measurment of
human eye movements. The eye-contact method is based on double
magnetic induction and allows for a lead-free eye coil. The major
characteristics of the method are a resolution of 8' of arc, a
linearity of up to 15 deg, and a frequency bandwidth of 3KHz. The
main advantages over methods based on signle magnetic induction are
pointed out.   Further improvement of resolution is discussed based
on theory and experiment. Results of measurments on human eye
movements are presented. We conclude that our new technique
considerably inproves eye movement measurment with eye-contact
methods, based on magnetic induction. The method is applicable to
man and animal.

File ID: 00085

Author: Ron, Samuel
Title: Target Velocity Based Prediction in Saccadic Vector
         Programming
Media: Vision Research vol 29, No. 9

Abstract/Citation:

ABSTRACT
   Two experiments have been designed to test whether the saccadic
system takes target motion into consideration in computing saccade
amplitude.   In one experiment, while the subject fixated straight
ahead, either a horizontal ramp-step-ramp or a horizontal
step-ramp target moved from left to right.   After the step, the
subject had to make a saccade and follow the target. In the second
set of experiments, the target, after an initial step, moved
extrafoveally from up to down at fixed velocity; a tone signaling
the subject to make a saccade to the target and follow it, was
delivered either after a variable delay (previewed condition) or
simultaneously with the initial target step (non-previewed
condition).   In both experiments, eye position at saccade end was
statistically different from target position 100 msec before saccade
onset only when the target slow motion was presented before the step
(i.e. in horizontal ramp-step-ramp and in previewed H-step V-ramp
paradigms), suggesting that target motion could be used by the
saccadic system to extrapolate the future target posistion, only if
the subject is given enough time to observe the target ramp motion
before the step.

File ID: 00086

Subject: Human Factors
Key Word 2:   Eye Tracking
Key Word 3:
Key Word 4:

Author: Schoessler, John P.
Title: Technical Note - A New Corrneal Electrode for
        Electroretinography
Media: Vision Research vol, 15.

Publication Date: 1-Jan-75
Publisher, City: Pergamon Press, Great Britain
Page #: 299-301
Document ID:
Agency: Vision Research
Type of Document: paper

Abstract/Citation:
Electroretinography is most ofter carried out through the use of a
fluid haptic lens with an implanted electrode. Several lenses of
this type have been described (Riggs, 1941; Burian 1953; Jacobson
1955; Karpe, 1961). There are a number of disadvantages in using a
Burian-Allen or Riggs type of eletrode, however. These include
discomfort, difficulty in maintaining fluid beneath the lens,
difficulty in inserting the removing the lens on many patients
(especially children), and the need for several different sizes of
haptic shells. The use of some of these device also may result in
considerable corneal damage (dawson, Zimmerman and Houde, 1974) and
it is generally reocmmended that corneal anesthetics be sued before
applying the ERG electrode (Karpe, 1961).

File ID: 00087

Subject: Human Factors
Key Word 2:   Keyboards
Key Word 3:
Key Word 4:

Author: Norman, Donald A.
Title: The DVORAK Revival: Is It Really Worth the Cost?

Media:

Publication Date: 23-Aug-83
Publisher, City:
Page #:
Document ID: Cntrct # N00014-79-C-0323, NR 667-37
Agency: Office of Naval research
Type of Document: brochure

Abstract/Citation:
The newspaper clipping printed along with this article (Lamb, 1983)
tells all; the standard, QWERTY keyboard has met its fate; the
Dvorak simplified keyboard (DSK) will (finally) take over.
 Personally, I doubt it. The studies in my laboratory indicate that
although DSK is indeed superior to the QWERTY keyboard, the benefits
do not outwiegh the costs of conversion to a new standard.   Of
course, it is always possible that a radical restructuring of the
typing task and keyboard input will yield significant benefits.   So
far, there are no proven prospects but many have hopes.
    .....There are several questions to be asked in reconsidering how
we type:
   How much gain will there be in type speed?
   How much faster can people learn it?
   How much less fatigue will there be?
   Is there anything better?
   What will happen to everyone who is now proficient at the QWERTY
keyboard?

File ID: 00088

Subject: Human Factors
Key Word 2:   Keyboards
Key Word 3:
Key Word 4:

Author: Norman, Donald A.
Title: Why Alphabetic Keyboards Are Not Easy to Use:
         Keyboard Layout Doesn't Much Matter
Media: Human Factors Society 24(5)

Publication Date: 1-Jan-82
Publisher, City:
Page #:   509-519
Document ID: 24(5)
Agency:
Type of Document: article

Abstract/Citation:
ABSTRACT
   These studies demonstrate the inferiority of alphabetically
organized keyboards as compared with a randomly organized keyboard
and the standard Sholes (qwerty) keyboard. Use of the alphabetic
keyboard requires considerable mental processing; the novice is
faced with a trade-off between mental processing and visual search,
and this makes different keyboard layouts equivalent. Comparison of
different keyboard layouts by computer simulation of expert typing
shows surprisingly little effect of keyboard arrangement for a wide
class of keyboards. Performance with some alphabetical layouts is
quite slow, but with others, it is within 2% of the speed achieved
when using the Sholes keyboard. Performance with the Dvorak
keyboard is only improved by about 5% over performance on the Sholes
keyboard. The conclusion is that it is not worth while to use
alphabetic keyboards for novice typisits, nor to change to the Dvorak
layout for experts. Keyboards can probably be improved, but only
through radical redesign of the present physical key configuration.

File ID: 00089

Subject: Human Factors
Key Word 2:   Keyboards
Key Word 3:
Key Word 4:

Author: Wlonk Enterprises
Title: Advertising Brochure

-Media:

Publication Date:
Publisher, City:
Page #:
Document ID: 1/1/89
Agency: Wlonk Enterprises
Type of Document: Brochure

Abstract/Citation:
The Wlonk Keyboard: Instructions

Note: A three page brochure which discusses a ten key keyboard and
how to use it. This is a chording keyboard i.e. a combination of
keys must be pressed to make one character.

File ID: 00090

Subject: Human Factors
Key Word 2:   Keyboards
Key Word 3:
Key Word 4:

Author: Volnak, William M.
Title: United States Patent

Media:

Publication Date: 7-Jul-87
Publisher, City: US Government, Washington D.C.
Page #:
Document ID: Patent # 4679030
Agency:
Type of Document: copy of patent

Abstract/Citation:

ABSTRACT
   A data entry device wherin key switches may be pressed singly and
in combination to generate a binary number corresponding to specific
characters and character combinations. A preferred embodiment of
the invention contemplates ten key switches, each key switch
corresponding to a different terminating member of an operator's
hand, such that a 10 digit binary number is generated. Other
embodiments of the present invention comtemplate a programmable
keyboard wherin characters and character combinations generated by
user operation of the keyboard are stored in a portable memory for
later access by the operator. Additionally, the device may be
operated in reverse such that binary numbers corresponding to
characters and character coombinations may be converted to patters
of tactile sensations perceived as characters and character
combinations.

CHAPTER NINE

ASTRONAUTICS CORPORATION OF AMERICA
ADCACS RESEARCH ACTIVITIES REPORT

Please note that Chapter 9 contains proprietary and trade secret information of Astronautics Corporation of America.

9-1.0   Zero Motion Master (ZMM) Arm

The ZMM arm provided by Astronautics (a proprietary technology used with permission) has been implemented into the prototype virtual workstation emulator environment.

Presently,, ZMM technology is the primary master technology being explored for three reasons. First, it is available and supported by Astronautics. Second, there is limited space in the Cupola module and full range motion may not be available. Third, the ZMM has proven force feedback. This feedback may be sufficient and is simplistic so that its cost is low, it is reliable, and it is easy to maintain.

The ZMM arm now utilizes 5 degrees-of-freedom on the arm (i.e. Shoulder Flexion/Extension, Shoulder Abduction/Adduction, Upper Arm Rotation, Elbow Flexion/Extension, and Forearm Rotation) with near future goals of Wrist Flexion/Extension and Wrist Abduction/Adduction. The current configuration consists of the ZMM arm as an input device which sends an analog signal to a A/D board and the interpretation software in a 286-based personal computer (PC2). The PC2 is reading a digital signal from the current 5 degrees-of-freedom on the arm and maps it to a range of motion for the virtual arm. This information is then sent across a distributed network to the graphics workstation via Internet Domain Stream Sockets. This data is read by the graphics terminal and is used to adjust the position of the virtual human arm that is graphically portrayed.

The interpretation that occurs at PC2 consists of scaling the digital output of each channel to a range of motion. This is accomplished through the utilization of a physiological data base called Ergobase by Biomechanics Corporation of America. This data base contains the range of motion for each degree-of-freedom that is addressed in the ZMM. Ergobase contains the ranges of motion in degrees for the above specified movements for the central 95 percentile of the population; by linearly transforming the digital output of the ZMM to an angle degree that is within the human range of motion for the specific joint, a truly anthropomorphic graphical arm in the virtual world results.

The virtual workstation as a whole is created with optimum anthropomorphic and human factor considerations at every step of its design.

Design of virtual scenario presentation will be dependent on calculations and analysis of optimal human range of motion.

In this new design, a fundamental decision must be made - are there more advantages in a force-based master or can a position-based master be implemented in

conjunction for best results? Another aspect in this decision is the fact that there are multiple position-based masters on the market that could be implemented quickly into the experimental sub-environments. Positive aspects of the forced-based system include the possibility of implementation of force feedback to the user. One more advantage would be that the hand and wrist would not have to move to perform the necessary functions and smaller areas of workspace are possible although this is not as strong a consideration for a ZMM hand as it is for the ZMM arm.

The use of the ZMM requires that both arms be strapped to the system. Ease and speed of entrance to and exit from the ZMM, as well as psychological and physiological effects of being bound by the ZMM are being considered.

9-2.0 Position-Based Master (PBM) Hand

The position-based master (PBM) hand provided by Astronautics has been successfully added to the fingers/hand/arm position data processing microcomputer to complement the ZMM arm. Although a ZMM hand apparatus is under continuing development, it will require more time to implement. It is also possible that a PBM hand is appropriate for the virtual command workstation. Position-based experimentation can be performed while engineering testing and independent experimentation on the ZMM hand are completed.

Each joint of the PBM hand has an associated potentiometer (pot). The resulting voltage signal from each pot (some limited range within 0 to 5 volts) is sent through a two stage operational amplifier circuit (Figure 5). The first stage isolates the signal from the rest of the circuit. This is useful since it essentially eliminates any effects of un-matched pot and wire resistances in the PBM. The second stage has two adjustable pots. One is for gain and the other is for level shifting so that the resulting voltage ranges 0 to 5 volts. The output of the second stage is properly conditioned for the full range of the analog-to-digital converter.

An Accukey keyboard could be integrated into the hand mechanism. The keyboard has only four keys per hand. In this way the operator does not need to unstrap himself from the input positioning system to enter character data. Crew preference and training on the keyboard needs to be tested.

9-3.0   Integration of ZMM to the Network system
(This work was performed at Marquette University ACT laboratory)

Ethernet has a bandwidth of 10 Mbits/sec. Considering network overhead, the useful bandwidth for the protopype emulator is estimated to be about 300 k-bytes/sec which has been determined to be sufficient for the protopype emulator. The estimated maximum bandwidth necessary for each I/O device is presented below.

a)   The ZMM arm has 7 degrees-of-freedom with 2 bytes necessary for each degree. The maximum update rate for graphical presentation is 60 Hz. There will be a maximum of two arms. Bandwidth = 2 arms * 7 degrees * 2 bytes * 60 Hz = 1.680 k-bytes/sec.

b)   The ZMM hand will have no more than 15 degrees-of-freedom. Bandwidth = 2 hands * 15 degrees * 2 bytes * 60 Hz = 3.600 k-bytes/sec.

c)   The maximum typing rate will probably not exceed 80 words/minute at 5 characters/word. If the ZMM hands are used to enter the character data, the larger number (this or item 2 above) should be used. If a separate keyboard is used, both bandwidths need to be considered. Bandwidth = 80 * 5 / 60 = 7 bytes/sec.

d)   Voice commands are estimated to occur less than once per second. Vocabularies greater than 256 commands are unlikely. Bandwidth = 1 byte/sec.

Appendix A   Graphic Program Listings

```c
/*
 *      This program moves the arm in a rectangle around the screen.  By
 *      varying the size of the rectangle different areas of the screen
 *      can be repeatedly touched by the hand.  This is used for testing
 *      various background scenarios and arm drawing subroutines.
 *
 *      It is assumed that the client is run on the Silicon Graphics
 *      Personal IRIS and the server is run on an IBM PC.  Therefore,
 *      the client reverses the byte sex.
 *
 *    % cc zx.c -o zx
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <sex.h>
/* Listen for connect requests on this arbitrarily-chosen port */
#define SERVER_PORT      5001

#define LINE_LEN                 80
#define DEGREES_OF_FREEDOM       32


main()
{
    float a[DEGREES_OF_FREEDOM];
    int b, i, j, sock, length, msgsock, cnt;
    char line[LINE_LEN];
    short int dof[DEGREES_OF_FREEDOM];
    struct sockaddr_in sin;

/* Create a socket */

    if ((sock = socket (AF_INET,SOCK_STREAM,0)) < 0) {
        perror("opening stream socket");
        exit(1);
    }

/* Initialize the socket's address structure */

    sin.sin_family = AF_INET;
    sin.sin_addr.s_addr = INADDR_ANY;
    sin.sin_port = htons(SERVER_PORT);

/* Assign an address to this socket */

    if (bind (sock,&sin,sizeof(sin)) < 0) {
        perror("binding stream socket");
        exit(1);
    }

    printf("Listening for connect requests on port %d\n", SERVER_PORT);

/* Prepare the socket queue for connection requests */

    listen(sock,5);
```

```c
        length = sizeof(sin);
        msgsock = accept(sock, &sin, &length);
        if (msgsock < 0) {
            perror("accept");
            exit(1);
        }

        printf("Connection from host %s, port %u\n",
            inet_ntoa(sin.sin_addr), ntohs(sin.sin_port));
        fflush(stdout);

/* Zero dof array. */
        for (i = 0;  i < DEGREES_OF_FREEDOM;  i++) {
            a[i] = 2.0;
            dof[i] = 0;}

/* set up the hand to be pointing */
            dof[7] = -60;
            dof[8] = 0;
            dof[9] = -60;
            dof[10] = -60;

            dof[11] = -60;
            dof[12] = 0;
            dof[13] = -60;
            dof[14] = -60;

            dof[15] = -60;
            dof[16] = 0;
            dof[17] = -60;
            dof[18] = -60;

            dof[19] = 0;
            dof[20] = 0;
            dof[21] = 0;
            dof[22] = 0;

            dof[23] = 0;
            dof[24] = 0;
            dof[25] = -40;
            dof[26] = -40;

for (i=7;i<DEGREES_OF_FREEDOM;i++) {
            a[i] = a[i]*(-1);}
        dof[1] = 119.0;
        dof[0] = 29.0;
        b = 1;

/* Read the syncronizing request for data from the client. */
        while ((cnt = read(msgsock, line, LINE_LEN)) > 0) {
            write(1, line, cnt);

/* move the arm in a square around the screen */
            if (b == 1) {
                dof[0] += a[0];
                if (dof[0] >= 30.0) a[0] = a[0]*(-1);
                if (dof[0] <= -30.0) a[0] = a[0]*(-1);
                if (dof[0] >= 30 || dof[0] <= -30.0) b = 2;}

            if (b == 2) {
```

```c
        dof[1] += a[1];
        if (dof[1] >= 120.0) a[1] = a[1]*(-1);
        if (dof[1] <= 60.0) a[1] = a[1]*(-1);
        if (dof[1] >= 120.0 || dof[1] <= 60.0) b = 1;}

    for (i=0;i<DEGREES_OF_FREEDOM;i++) {
      dof[i] = swap_half(dof[i]);}

/* Write the degrees of freedom array to the socket. */
    write(msgsock, dof, sizeof(dof));


    for (i=0;i<DEGREES_OF_FREEDOM;i++) {
      dof[i] = swap_half(dof[i]);}

  }

  close(msgsock);
  printf("Connection closed\n");
}
```

```c
    if (msgsock < 0) {
        perror("accept");
        exit(1);
    }

    printf("Connection from host %s, port %u\n",
        inet_ntoa(sin.sin_addr), ntohs(sin.sin_port));
    fflush(stdout);

/* Zero dof array. */
    for (i = 0;   i < DEGREES_OF_FREEDOM;   i++) {
        dof[i] = 0;}

/* 90 degrees sets the arm up moving straight away from the user */
/* when the cursor is in the center of the screen and shoulder.  */
    dof[1] = 90.0;

/* Read the syncronizing request for data from the client. */
    while ((cnt = read(msgsock, line, LINE_LEN)) > 0) {
        write(1, line, cnt);

        for (i=0;i<DEGREES_OF_FREEDOM;i++) {
           dof[i] = swap_half(dof[i]);}

/* Write the degrees of freedom array to the socket. */
        write(msgsock, dof, sizeof(dof));


        for (i=0;i<DEGREES_OF_FREEDOM;i++) {
           dof[i] = swap_half(dof[i]);}
    }

    close(msgsock);
    printf("Connection closed\n");
}
```

```c
/*
 *      This program will continuously increase and decrease the angles
 *      in the arm and hand to allow testing and debugging of various
 *      background scenarios and arm drawing subroutines.
 *
 *      It is assumed that the client is run on the Silicon Graphics
 *      Personal IRIS and the server is run on an IBM PC.  Therefore,
 *      the client reverses the byte sex.
 *
 *    TO COMPILE: % cc zz.c -o zz
 *    TO EXECUTE: % zz <RETURN>
 */

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <sex.h>
/* Listen for connect requests on this arbitrarily-chosen port */
#define SERVER_PORT      5001

#define LINE_LEN                80
#define DEGREES_OF_FREEDOM      32

main()
{
    float a[DEGREES_OF_FREEDOM];
    int i, j, sock, length, msgsock, cnt;
    char line[LINE_LEN];
    short int dof[DEGREES_OF_FREEDOM];
    struct sockaddr_in sin;

/* Create a socket */

    if ((sock = socket (AF_INET,SOCK_STREAM,0)) < 0) {
        perror("opening stream socket");
        exit(1);
    }

/* Initialize the socket's address structure */

    sin.sin_family = AF_INET;
    sin.sin_addr.s_addr = INADDR_ANY;
    sin.sin_port = htons(SERVER_PORT);

/* Assign an address to this socket */

    if (bind (sock,&sin,sizeof(sin)) < 0) {
        perror("binding stream socket");
        exit(1);
    }

    printf("Listening for connect requests on port %d\n", SERVER_PORT);

/* Prepare the socket queue for connection requests */

    listen(sock,5);
```

```c
    length = sizeof(sin);
    msgsock = accept(sock,  &sin,  &length);
    if (msgsock < 0) {
        perror("accept");
        exit(1);
    }

    printf("Connection from host %s, port %u\n",
        inet_ntoa(sin.sin_addr), ntohs(sin.sin_port));
    fflush(stdout);

/* Zero dof array. */
    for (i = 0;   i < DEGREES_OF_FREEDOM;   i++) {
        a[i] = 1.0;
        dof[i] = 0;}

    for (i=7;i<DEGREES_OF_FREEDOM;i++) {
        a[i] = a[i]*(-1);}
    dof[1] = 90.0;

/* Read the syncronizing request for data from the client. */
    while ((cnt = read(msgsock, line, LINE_LEN)) > 0) {
        write(1, line, cnt);

/* increase or decrease the angles in the DOF array */

/* arm, elbow and wrist angles */
        dof[0] += a[0];
        if (dof[0] >= 40.0) a[0] = a[0]*(-1);
        if (dof[0] <= -40.0) a[0] = a[0]*(-1);

        dof[1] += a[1];
        if (dof[1] >= 100.0) a[1] = a[1]*(-1);
        if (dof[1] <= 30.0) a[1] = a[1]*(-1);

        dof[2] += a[2];
        if (dof[2] >= 30.0) a[2] = a[2]*(-1);
        if (dof[2] <= 0) a[2] = a[2]*(-1);

        dof[3] += a[3];
        if (dof[3] >= 45.0) a[3] = a[3]*(-1);
        if (dof[3] <= 0) a[3] = a[3]*(-1);

        dof[4] += a[4];
        if (dof[4] >= 40.0) a[4] = a[4]*(-1);
        if (dof[4] <= -40.0) a[4] = a[4]*(-1);

        dof[4] += a[4];
        if (dof[4] >= 50.0) a[4] = a[4]*(-1);
        if (dof[4] <= 0) a[4] = a[4]*(-1);

        dof[5] += a[5];
        if (dof[5] >= 40.0) a[5] = a[5]*(-1);
        if (dof[5] <= 0) a[5] = a[5]*(-1);

        dof[6] += a[6];
        if (dof[6] >= 30.0) a[6] = a[6]*(-1);
        if (dof[6] <= 0) a[6] = a[6]*(-1);

/* finger angles */
```

```c
        for (i=0;i<5;i++){
           for (j=0;j<4;j++){
              dof[7+i*4+j] += a[7+i*4+j]/10.;
              if (dof[7+i*4+j] >= 0.0) a[7+i*4+j] = a[7+i*4+j]*(-1);
              if (dof[7+i*4+j] <= -30.0)    a[7+i*4+j] = a[7+i*4+j]*(-1);

}}

        for (i=0;i<DEGREES_OF_FREEDOM;i++) {
           dof[i] = swap_half(dof[i]);}

/* Write the degrees of freedom array to the socket. */
        write(msgsock, dof, sizeof(dof));


        for (i=0;i<DEGREES_OF_FREEDOM;i++) {
           dof[i] = swap_half(dof[i]);}
        }

    close(msgsock);
    printf("Connection closed\n");
}
```

```
/*************************************************************/
/*          Jeffrey J. Garside                               */
/*          ADCACS Project                                   */
/*          June, 1991                                       */
/*          Program:  colorbox.c                             */
/*          Features:  Box Drawing and Manipulation          */
/*************************************************************/

/* TO COMPILE:   % cc colorbox.c -lm -lgl_s -lc_s -s -o colorbox <RET> */
/* TO EXECUTE:   % colorbox iris <RETURN>    OR   % colorbox pc2 <RET> */

#include <math.h>
#include <gl/gl.h>
#include <gl/device.h>
#define SIN 0
#define COS 1
#define _BSD_SIGNALS
#define SIDES 8    /* number of sides to polygons making up arm & fingers */
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <sex.h>
#define LINE_LEN                80
#define DEGREES_OF_FREEDOM      32
/* The server listens for connect requests on this port */
#define SERVER_PORT     5001

static int die();

Matrix Identity = { 1, 0, 0, 0,  0, 1, 0, 0,  0, 0, 1, 0,  0, 0, 0, 1 };

float mat[] = {
        AMBIENT, .1, .1, .1,
/*      DIFFUSE, .996, .508, .391,*/
        DIFFUSE, .999, .400, .240,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
static float lm[] = {
        AMBIENT, .1, .1, .1,
        LOCALVIEWER, 0,
        LMNULL};
static float lt[] = {
        LCOLOR, 1., 1., 1.,
        POSITION, .5, .5, 1., 0.,
        LMNULL};

float mat2[] = {
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1., 0., 0.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat3[] = {
        AMBIENT, .1, .1, .1,
```

```c
            DIFFUSE, 1., 1., 0.,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
    };

    float mat4[] = {
            AMBIENT, .1, .1, .1,
            DIFFUSE, 0., 0., 1.,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
    };

    float mat5[] = {
            AMBIENT, .1, .1, .1,
            DIFFUSE, 0., 0.666, 0.035,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
    };


    static float mats[5][4][4] =
            {{{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}}};

    float angle[SIDES][2];

    main(argc,argv)
        int argc;
        char **argv;
    {
        int i, cnt, sock;
        struct sockaddr_in sin;
        struct hostent *hp;
        char line[LINE_LEN];
        short dof[DEGREES_OF_FREEDOM];


long xorigin, yorigin, xsize, ysize;
float rx, ry;
```

```c
    short val;

    for (i=0;i<32;i++) {
        dof[i] = 0;}

    if (argc != 2) {
        printf("usage: %s host\n", argv[0]);
        exit(1);}

    if ((sock = socket (AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("Can't open socket");
        exit(1);}

    /* Initialize the socket address to the server's address. */

    bzero((char *) &sin, sizeof(sin));
    sin.sin_family = AF_INET;
    hp = gethostbyname(argv[1]);      /* to get host address */
    if (hp == NULL) {
        herror(argv[1]);
        exit(1);}
    bcopy (hp->h_addr, &(sin.sin_addr.s_addr), hp->h_length);
    sin.sin_port = htons(SERVER_PORT);

    /* Connect to the server. */

    if (connect(sock, &sin,sizeof(sin)) < 0) {
        close(sock);
        perror("Connect to server");
        exit(1);}
    printf("Connection established. The data is entered at the server.\n");

    /*
     * If the server goes away while sending data, we'll get a
     * SIGPIPE signal.  Catch it so we can print an error message.
     */
    (void) signal(SIGPIPE, die);

        prefsize(1280, 1024);
        defineangle();
        noborder();
        winopen("arm");
        qdevice(ESCKEY);
        getorigin(&xorigin, &yorigin);
        getsize(&xsize, &ysize);
        doublebuffer();
        RGBmode();
        gconfig();
        backface(TRUE);
        lsetdepth(0, 0x7fffff);
        zbuffer(1);
        mmode(MVIEWING);
        loadmatrix(Identity);
        perspective(700, xsize/(float)ysize, 50., 300.0); /* was .25,300.0 */
        lmdef(DEFMATERIAL, 1, 0, mat);
        lmdef(DEFMATERIAL, 2, 0, mat2);
        lmdef(DEFMATERIAL, 3, 0, mat3);
        lmdef(DEFMATERIAL, 4, 0, mat4);
        lmdef(DEFMATERIAL, 5, 0, mat5);
        lmdef(DEFLIGHT, 1, 0, lt);
```

```c
        lmdef(DEFLMODEL, 1, 0, lm);
        lmbind(MATERIAL, 1);
        lmbind(LMODEL, 1);
        lmbind(LIGHT0, 1);
        translate(0, 0, -150); /* was -150 */
/* remove this line when not using demo
        rotate(1800,'y');*/
        while (!(qtest() && qread(&val) == ESCKEY && val == 0)) {

            /* Send a synchronizing request for data to the server. */
            strcpy(line, "?");
            cnt = strlen(line);
            if (write(sock, line, cnt) < 0) {
              perror("Error writing to socket.");
              exit(1);}

                    czclear(0xffffff,0x7fffff);
                    lmbind(MATERIAL,1);
                    translate(0,0,50.);
                    drawarm(dof,mats);
                    translate(0,0,-50.);

/*                  lmbind(MATERIAL,2);*/
                    clldet(mats);
                    swapbuffers();

            /* Read the degrees of freedom array from the socket.*/
            if (read(sock, dof, sizeof(dof)) != sizeof(dof)) {
              printf("Error reading data from server.\n");
              exit(1);}

            /* Reverse the byte sex. */
            for (i = 0;  i < DEGREES_OF_FREEDOM;  i++) {
              dof[i] = swap_half(dof[i]);}

/*          for (i=0;i<DEGREES_OF_FREEDOM;i++) {
                  printf(" %d",dof[i]);}
            printf("\n");
*/
        }

        printf ("Done\n");
        exit(0);

}

clldet(float mats[5][4][4])
{
int i,j,k,size;
float n[3];
static float va[][3] = {
        {1.,   -1.,    0.},
        {1.,    1.,    0.},
        {-1., -1.,    0.},
        {-1.,  1.,    0.},
        {1.,   -1.,  -250.},
        {1.,    1.,  -250.},
        {-1., -1.,  -250.},
        {-1.,  1.,  -250.}
};
```

```
static float vb[][3] = {
        {1.,    -1.,    0.},
        {1.,     1.,    0.},
        {-1.,   -1.,    0.},
        {-1.,    1.,    0.},
        {1.,    -1.,   -250.},
        {1.,     1.,   -250.},
        {-1.,   -1.,   -250.},
        {-1.,    1.,   -250.}
};
        size = 45; /* change this to change box size */
        for (i = -1;i < 2;i = i + 2) {
         for (j = -1;j < 2;j = j + 2) {
          for (k=0;k<8;k++) {
             va[k][0] =   80*i + size*vb[k][0];
             va[k][1] =   80*j + size*vb[k][1];}
        if (i == -1 && j == -1) lmbind(MATERIAL,2);
        if (i == -1 && j == 1) lmbind(MATERIAL,3);
        if (i == 1 && j == -1) lmbind(MATERIAL,5);
        if (i == 1 && j == 1) lmbind(MATERIAL,4);

        n[0] =   0.;  n[1] =  0.;   n[2] =   1.;
        n3f(n);
        bgnqstrip();
        v3f(va[0]);
        v3f(va[1]);
        v3f(va[2]);
        v3f(va[3]);
        endqstrip();

        n[0] =   0.;  n[1] =  1.;   n[2] =   0.;
        n3f(n);
        bgnqstrip();
        v3f(va[1]);
        v3f(va[5]);
        v3f(va[3]);
        v3f(va[7]);
        endqstrip();

        n[0] =   0.;  n[1] =  0.;   n[2] = -1.;
        n3f(n);
        bgnqstrip();
        v3f(va[5]);
        v3f(va[4]);
        v3f(va[7]);
        v3f(va[6]);
        endqstrip();

        n[0] =   0.;  n[1] = -1.;   n[2] =   0.;
        n3f(n);
        bgnqstrip();
        v3f(va[4]);
        v3f(va[0]);
        v3f(va[6]);
        v3f(va[2]);
        endqstrip();

        n[0] =   1.;  n[1] =  0.;   n[2] =   0.;
        n3f(n);
        bgnqstrip();
```

```
            v3f(va[4]);
            v3f(va[5]);
            v3f(va[0]);
            v3f(va[1]);
            endqstrip();


            n[0] =  -1.;  n[1] =  0.;  n[2] =  0.;
            n3f(n);
            bgnqstrip();
            v3f(va[2]);
            v3f(va[3]);
            v3f(va[6]);
            v3f(va[7]);
            endqstrip();}}

            mats[3][3][3] += 1.;
            mats[3][3][3] -= 1.;
}


defineangle()
{
float theta, dtheta = 2.*M_PI/SIDES;
int i;
            for (i = 0, theta = 0.;  i < SIDES;  i++, theta += dtheta)  {
                    angle[i][SIN] = sin(theta);
                    angle[i][COS] = cos(theta);
}}

static int die()
{
    fprintf(stderr, "Server closed connection\n");
    exit(1);
}

drawarm(short int dof[DEGREES_OF_FREEDOM], float mats[5][4][4])
{

/* all dimensions are measured in centimeters (cm) */
static float ftr[5][3] = {    /* finger translations [finger #][x,y,z]*/
          {2.6,0.0,0.0},
          {1.0,0.0,0.0},
          {-.8,0.0,0.0},
          {-2.5,0.0,0.0},
          {-1.8,-6.0,0.0}};

static float aang[7] = {
          0,1800,450,450,900,300,300};

static float fang[5][4] =
          {{-200,0,-200,-200},
           {-200,0,-200,-200},
           {-200,0,-200,-200},
           {-200,0,-200,-200},
           {-200,0,-200,-200}};

static float delta = {.01};

static float frad[5][4][3] = {
```

```c
/* finger cylinder radii and heights [finger #][digit #][x,y,z] */
        {{1.0,1.8,.8},
         {.95,1.8,.75},
         {.85,1.8,.6},
         {.85,1.5,.55}},           /* pinky */

        {{1.15,2.5,1.15},
         {1.05,2.5,.85},
         {.95,2.0,.75},
         {.9,1.2,.65}},            /* ring finger */

        {{1.1,3.0,1.1},
         {1.1,2.7,1.0},
         {.95,2.2,.8},
        - {.95,1.5,.7}},           /* middle finger */

        {{.85,2.4,.85},
         {.85,2.2,.85},
         {.8,2.0,.75},
         {.75,1.3,.65}},           /* index finger */

        {{1.5,4.0,1.5},
         {1.1,2.5,1.25},
         {1.1,2.0,.9},
         {.85,1.0,.6}}};           /* thumb */

static float sch = { 10.0 };       /* shoulder cap height */

static float arad[4][3] = {        /* arm cylinder radii and heights [digit #][x,y,z
        {6.0,30.0,6.0}, /* was 6,30,6 */
        {5.0,25.0,4.5}, /* was 5,25,4.5 */
        {3.0,8.0,3.0},
        {4.0,1.15,1.2}};

static float tang[3] = { 450, -900, 450 };  /* thumb rotation angles x, y, and z

float x, y, z, dz;
float n[3], v[3];
int i,j, da;
        pushmatrix();
        armang(fang,aang,dof);
/* draw shoulder */
        rotate(aang[1],'x');
        rotate(aang[0],'z');
        drawcap(arad[0][0],sch,arad[0][2]);                /* cap */
/* draw upper arm */
        rotate(1800,'x');
        translate(0,-.01,0);
        drawcyl(arad[0][0],arad[0][2],arad[1][0],arad[1][2],arad[0][1]);/* cylin
/* draw elbow */
        translate(0,arad[0][1]-delta,0);
        drawcap(arad[1][0],arad[1][0],arad[1][2]);         /* elbow nuckle */
        rotate(aang[3],'x');
        rotate(aang[2],'z');
/* draw lower arm */
        drawcyl(arad[1][0],arad[1][2],arad[2][0],arad[2][2],arad[1][1]);/* cylin
/* draw wrist */
        translate(0,arad[1][1]-delta,0);
        drawcap(arad[2][0],arad[2][0],arad[2][2]);  /* wrist nuckle */
        rotate(aang[4],'y');
```

```
        rotate(aang[5],'x');
        rotate(aang[6],'z');
/* draw hand */
        drawcyl(arad[2][0],arad[2][2],arad[3][0],arad[3][2],arad[2][1]);/*cylind
/* cap hand stump */
        translate(0,arad[2][1]-delta,0);
        drawcap(arad[3][0],arad[3][1],arad[3][2]);                      /* cap */

/* draw all 5 fingers */
     for (i=0;i<5;i++) {
        pushmatrix();
        translate(ftr[i][0],ftr[i][1],ftr[i][2]);
        if (i==4) {                    /* thumb */
                rotate(tang[1],'y');
                rotate(tang[0],'x');                /* remove numbers later */
                rotate(tang[2],'z');
                }
        rotate(fang[i][0],'x');
        rotate(fang[i][1],'z');
        drawcyl(frad[i][0][0],frad[i][0][2],frad[i][1][0],frad[i][1][2],frad[i][
        translate(0,frad[i][0][1],0);
        drawnuckle(frad[i][1][0],frad[i][1][2],-fang[i][2]);   /* nuckle */
        rotate(fang[i][2],'x');
        drawcyl(frad[i][1][0],frad[i][1][2],frad[i][2][0],frad[i][2][2],frad[i][
        translate(0,frad[i][1][1],0);
        drawnuckle(frad[i][2][0],frad[i][2][2],-fang[i][3]);   /*nuckle*/
        rotate(fang[i][3],'x');
        drawcyl(frad[i][2][0],frad[i][2][2],frad[i][3][0],frad[i][3][2],frad[i][
        translate(0,frad[i][2][1],0);

        fingertip1(1, i, frad, mats);/*frad[i][3][0],frad[i][3][1],frad[i][3][2]

/*      drawcap(frad[i][3][0],frad[i][3][1],frad[i][3][2]); cap */

        popmatrix();
     }
        popmatrix();
}

fingertip1(int d, int i, float frad[5][4][3], float mats[5][4][4])

/*d=1 if calling from drawing subroutine,
  d=0 if calling from collision detection subroutine*/

{
        int j,k;

        static float m[4][4] =
        {{0.0,0.0,0.0,0.0},
         {0.0,0.0,0.0,0.0},
         {0.0,0.0,0.0,0.0},
         {0.0,0.0,0.0,0.0}};

        pushmatrix();

        if (d == 1)
          getmatrix(m);

        if (d == 0)
          loadmatrix(mats[i][4][4]);
```

```
            for (j=0;j<4;j++) {
               for (k=0;k<4;k++) {
                   mats[i][j][k]=m[j][k];}}

            drawcap(frad[i][3][0],frad[i][3][1],frad[i][3][2]);

            popmatrix();
}
armang(float fng[5][4], float ang[7], short int dof[DEGREES_OF_FREEDOM])

{
int i,j;

        for (i=0;i<7;i++) {
            ang[i] = 10*(float)dof[i];}

        for (i=0;i<5;i++) {
           for (j=0;j<4;j++) {
               fng[i][j] = 10*(float)dof[7+i*4+j];}}   /* 10 can be removed */

}
drawcyl(rx1,rz1,rx2,rz2,dy)
float rx1,rz1,rx2,rz2,dy;
{
float x, y, z;
float n[3], v[3];
int j;

        bgnqstrip();
        for (j = 0;   j <= SIDES;   j++)   {
                if (j == SIDES) {
                        x = angle[0][COS];
                        z = angle[0][SIN];
                }
                else {
                        x = angle[j][COS];
                        z = angle[j][SIN];
                }
                n[0] = x;   n[1] = 0;   n[2] = z;
                n3f(n);
                v[0] = rx1*x;   v[1] = 0.;   v[2] = rz1*z;
                v3f(v);
                v[0] = rx2*x;   v[1] = dy;   v[2] = rz2*z;
                v3f(v);
        }
        endqstrip();
}

drawcap(rx,ry,rz)
float rx,ry,rz;
{
float cosphi, sinphi, cospdp, sinpdp, costheta, sintheta;
float x, y, z;
float n[3], v[3];
int i, j;

        for (i = 0;  i < 4;  i++)   {
```

```
                cosphi = angle[i][COS];
                sinphi = angle[i][SIN];
                cospdp = angle[i+1][COS];
                sinpdp = angle[i+1][SIN];
                bgnqstrip();
                for (j = 0;   j <= SIDES;  j++)   {
                        if (j == SIDES) {
                                costheta = angle[0][COS];
                                sintheta = angle[0][SIN];
                        }
                        else {
                                costheta = angle[j][COS];
                                sintheta = angle[j][SIN];
                        }
                        x = costheta * cosphi;
                        y =             sinphi;
                        z = sintheta * cosphi;
                        n[0] = x;   n[1] = y;   n[2] = z;
                        n3f(n);
                        v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                        v3f(v);
                        x = costheta * cospdp;
                        y =             sinpdp;
                        z = sintheta * cospdp;
                        n[0] = x;   n[1] = y;   n[2] = z;
                        n3f(n);
                        v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                        v3f(v);
                }
                endqstrip();
        }

        bgntmesh();
        n[0] = 0.;   n[1] = 1.;   n[2] = 0.;
        n3f(n);
        v[0] = 0.;   v[1] = ry;   v[2] = 0.;
        v3f(v);
        for (j = SIDES;  j >= 0;  j--)   {
                if (j == SIDES) {
                        x = angle[0][COS] * cospdp;
                        y =                  sinpdp;
                        z = angle[0][SIN] * cospdp;
                }
                else {
                        x = angle[j][COS] * cospdp;
                        y =                  sinpdp;
                        z = angle[j][SIN] * cospdp;
                }
                n[0] = x;   n[1] = y;   n[2] = z;
                n3f(n);
                v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                v3f(v);
                swaptmesh();
        }
        endtmesh();
}

drawnuckle(rz,rx,anc)
float rx,rz;  /* radius of shpere */
int  anc;  /* angle of next cylinder */
```

```c
{
float costheta, sintheta, costdt, sintdt, cosphi, sinphi;
float x, y, z;
float n[3], v[3];
int i, j, loop;

        if (anc == 0) return;
        rotate(-900,'y');
        if        (anc <= 180) loop = 1;
        else if (anc <= 360) loop = 2;
        else if (anc <= 540) loop = 3;
        else if (anc <= 720) loop = 4;
        else                   loop = 5;
        for (i = 0;  i < loop;  i++)  {
                costheta = angle[i][COS];
                sintheta = angle[i][SIN];
                costdt   = angle[i+1][COS];
                sintdt   = angle[i+1][SIN];
                bgnqstrip();
                for (j = -5;  j <= 5;  j++)  {
                        if (j < 0) {cosphi = angle[j+SIDES][COS];
                                    sinphi = angle[j+SIDES][SIN];}
                        else {      cosphi = angle[j][COS];
                                    sinphi = angle[j][SIN];}
                        x = cosphi * costheta;
                        y = cosphi * sintheta;
                        z = sinphi;
                        n[0] = x;  n[1] = y;  n[2] = z;
                        n3f(n);
                        v[0] = rx*x;  v[1] = rx*y;  v[2] = rz*z;
                        v3f(v);
                        if (i == loop-1) {
                                x = cosphi * cos(anc*2.*M_PI/3600.);
                                y = cosphi * sin(anc*2.*M_PI/3600.);
                                z = sinphi;
                                n[0] = x;  n[1] = y;  n[2] = z;
                                n3f(n);
                                v[0] = rx*x;  v[1] = rx*y;  v[2] = rz*z;
                                v3f(v);
                        }
                        else {
                                x = cosphi * costdt;
                                y = cosphi * sintdt;
                                z = sinphi;
                                n[0] = x;  n[1] = y;  n[2] = z;
                                n3f(n);
                                v[0] = rx*x;  v[1] = rx*y;  v[2] = rz*z;
                                v3f(v);
                        }
                }
                endqstrip();
        }
        rotate(900,'y');
}
```

```c
/**********************************************************/
/*          Jeffrey J. Garside                            */
/*          ADCACS Project                                */
/*          June, 1991                                    */
/*          Program:  background.c                        */
/*          Features:  Collision Detection,               */
/*                     and Box Manipulation               */
/**********************************************************/

/* TO COMPILE:   % cc background.c -lm -lgl_s -lc_s -s -o background <RET> */
/* TO EXECUTE:   % background iris <RETURN>    OR   % background pc2 <RET> */

#include <math.h>
#include <gl/gl.h>
#include <gl/device.h>
#define SIN 0
#define COS 1
#define _BSD_SIGNALS
#define SIDES 8    /* number of sides to polygons making up arm & fingers */
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <sex.h>
#define LINE_LEN                    80
#define DEGREES_OF_FREEDOM          32
/* The server listens for connect requests on this port */
#define SERVER_PORT        5001

static int die();

Matrix Identity = { 1, 0, 0, 0,  0, 1, 0, 0,  0, 0, 1, 0,  0, 0, 0, 1 };


static float lm[] = {
        AMBIENT, .1, .1, .1,
        LOCALVIEWER, 0,
        LMNULL};

static float lt0[] = {
        LCOLOR, 1., 1., 1.,
        POSITION, -0.4, 0.4, 1., 0.,   /* was .5, .5, 1., 0. */
        LMNULL};

static float lt1[] = {
        LCOLOR, 1., 1., 1.,
        POSITION, 0.4, -0.4, 1., 0.,   /* was .5, .5, 1., 0. */
        LMNULL};

float mat1[] = {   /* skin */
        AMBIENT, .1, .1, .1,
/*      DIFFUSE, .996, .508, .391,*/
        DIFFUSE, .999, .400, .240,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat2[] = {   /* blue */
```

```c
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, 0., 1.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat3[] = {   /* lt green */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, 1., 0.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat4[] = {   /* cyan */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 0.1, 1., 1.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat5[] = {   /* lt blue */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 0., 0.666, 0.1,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,

};

float mat6[] = {   /* yellow */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1., 1., 0.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat7[] = {   /* pale green */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .5, 1., 0.5,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat8[] = {   /* deep purple */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 0.5, 0., .5,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,
};

float mat9[] = {   /* purple */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1., 0., 1.0,
        SPECULAR, .5, .5, .5,
```

```c
            SHININESS, 0,
            LMNULL,
};
float mat10[] = {   /* black */
            AMBIENT, .1, .1, .1,
            DIFFUSE, .0, 0., .0,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
};
float mat11[] = {   /* white */
            AMBIENT, .1, .1, .1,
            DIFFUSE, 1.0, 1.0, 1.0,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
};
float mat100[] = {   /* red */
            AMBIENT, .1, .1, .1,
            DIFFUSE, 1.0, 0.0, 0.0,
            SPECULAR, .5, .5, .5,
            SHININESS, 0,
            LMNULL,
};

static double mats[5][4][4] =
            {{{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}},
             {{0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0},
              {0.0,0.0,0.0,0.0}}};

double angle[SIDES][2];

float ftloc[][3] = {
            {0.,0.,0.},
            {0.,0.,0.},
            {0.,0.,0.},
            {0.,0.,0.},
            {0.,0.,0.}};

main(argc,argv)
    int argc;
    char **argv;
{
```

```c
    int i, cnt, sock;
    struct sockaddr_in sin;
    struct hostent *hp;
    char line[LINE_LEN];
    short dof[DEGREES_OF_FREEDOM];


long xorigin, yorigin, xsize, ysize;
float rx, ry;
short val;

    for (i=0;i<32;i++) {
        dof[i] = 0;}

    if (argc != 2) {
        printf("usage: %s host\n", argv[0]);
        exit(1);}

    if ((sock = socket (AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("Can't open socket");
        exit(1);}

    /* Initialize the socket address to the server's address. */

    bzero((char *) &sin, sizeof(sin));
    sin.sin_family = AF_INET;
    hp = gethostbyname(argv[1]);       /* to get host address */
    if (hp == NULL) {
        herror(argv[1]);
        exit(1);}
    bcopy (hp->h_addr, &(sin.sin_addr.s_addr), hp->h_length);
    sin.sin_port = htons(SERVER_PORT);

    /* Connect to the server. */

    if (connect(sock, &sin,sizeof(sin)) < 0) {
        close(sock);
        perror("Connect to server");
        exit(1);}
    printf("Connection established. The data is entered at the server.\n");

    /*
     * If the server goes away while sending data, we'll get a
     * SIGPIPE signal.  Catch it so we can print an error message.
     */
    (void) signal(SIGPIPE, die);

        prefsize(1280, 1024);
        defineangle();
        noborder();
        winopen("arm");
        qdevice(ESCKEY);
        getorigin(&xorigin, &yorigin);
        getsize(&xsize, &ysize);
        doublebuffer();
        RGBmode();
        gconfig();
        backface(TRUE);
        lsetdepth(0, 0x7fffff);
        zbuffer(1);
```

```
mmode(MVIEWING);
loadmatrix(Identity);
perspective(600, xsize/(float)ysize, 50., 300.0); /* was .25,300.0 */
lmdef(DEFMATERIAL,  1,  0,  mat1);
lmdef(DEFMATERIAL,  2,  0,  mat2);
lmdef(DEFMATERIAL,  3,  0,  mat3);
lmdef(DEFMATERIAL,  4,  0,  mat4);
lmdef(DEFMATERIAL,  5,  0,  mat5);
lmdef(DEFMATERIAL,  6,  0,  mat6);
lmdef(DEFMATERIAL,  7,  0,  mat7);
lmdef(DEFMATERIAL,  8,  0,  mat8);
lmdef(DEFMATERIAL,  9,  0,  mat9);
lmdef(DEFMATERIAL,10,  0,  mat10);
lmdef(DEFMATERIAL,11,  0,  mat11);
lmdef(DEFMATERIAL,100,0,mat100);
lmdef(DEFLIGHT,  1,  0,  lt0);
lmdef(DEFLIGHT,  2,  0,  lt1);
lmdef(DEFLMODEL,  1,  0,  lm);
lmbind(MATERIAL, 1);
lmbind(LMODEL, 1);
lmbind(LIGHT0, 1);
lmbind(LIGHT1, 2);
translate(0, 0, -150); /* was -150 */
while (!(qtest() && qread(&val) == ESCKEY && val == 0)) {

  /* Send a synchronizing request for data to the server. */
  strcpy(line, "?");
  cnt = strlen(line);
  if (write(sock, line, cnt) < 0) {
    perror("Error writing to socket.");
    exit(1);}

        czclear(0x000000,0x7fffff);
        lmbind(MATERIAL,1);
        translate(0,0,60.);
        drawarm(dof,mats);
        translate(0,0,-60.);
        backgnd();
        swapbuffers();

  /* Read the degrees of freedom array from the socket.*/
  if (read(sock, dof, sizeof(dof)) != sizeof(dof)) {
    printf("Error reading data from server.\n");
    exit(1);}

  /* Reverse the byte sex. */
  for (i = 0;  i < DEGREES_OF_FREEDOM;  i++) {
    dof[i] = swap_half(dof[i]);}

   for (i=0;i<DEGREES_OF_FREEDOM;i++) {
        printf(" %d",dof[i]);}
   printf("\n");

}

printf ("Done\n");
exit(0);
```

```
        for (i = -6;i<7;i++) {
          b[0] = 10.*i;
          j = 11;
          for (k=0;k<5;k++) {
            if(      (ftloc[k][0] >= b[0] - b[3])
                 && (ftloc[k][0] <= b[0] + b[3])
                 && (ftloc[k][1] >= b[1] - b[4])
                 && (ftloc[k][1] <= b[1] + b[4])
                 && (ftloc[k][2]+150. >= b[2] - b[5])
                 && (ftloc[k][2]+150. <= b[2] + b[5]) )    j=100;}

          drawbox(b,j);}

        lmbind(MATERIAL,10);
        circfi(45.,45.,15.);
        circfi(-45.,45.,15.);
        for (i=0;i<14;i++) {
          lmbind(MATERIAL,(int)i/2+3);
          circfi(c[i][0],c[i][1],1.);
          circfi(-c[i][1],c[i][0],1.);}

}

drawbox(bd, mat_index)
float bd[6];       /* box dimensions          */
int   mat_index;  /* material of box (color) */
{
float n[3], va[8][3];

        lmbind(MATERIAL, mat_index);

        va[0][0] = va[1][0] = va[4][0] = va[5][0] =  bd[0] + bd[3];
        va[2][0] = va[3][0] = va[6][0] = va[7][0] =  bd[0] - bd[3];

        va[1][1] = va[3][1] = va[5][1] = va[7][1] =  bd[1] + bd[4];
        va[0][1] = va[2][1] = va[4][1] = va[6][1] =  bd[1] - bd[4];

        va[0][2] = va[1][2] = va[2][2] = va[3][2] =  bd[2] + bd[5];
        va[4][2] = va[5][2] = va[6][2] = va[7][2] =  bd[2] - bd[5];

        n[0] =  0.;  n[1] =  0.;  n[2] =  1.;  n3f(n);
        bgnqstrip();
        v3f(va[0]);  v3f(va[1]);  v3f(va[2]);  v3f(va[3]);
        endqstrip();

        n[0] =  0.;  n[1] =  1.;  n[2] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[1]);  v3f(va[5]);  v3f(va[3]);  v3f(va[7]);
        endqstrip();

        n[0] =  0.;  n[1] =  0.;  n[2] = -1.;  n3f(n);
        bgnqstrip();
        v3f(va[5]);  v3f(va[4]);  v3f(va[7]);  v3f(va[6]);
        endqstrip();

        n[0] =  0.;  n[1] = -1.;  n[2] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[4]);  v3f(va[0]);  v3f(va[6]);  v3f(va[2]);
        endqstrip();
```

```
        n[0] =   1.;   n[1] =   0.;   n[2] =   0.;   n3f(n);
        bgnqstrip();
        v3f(va[4]);   v3f(va[5]);   v3f(va[0]);   v3f(va[1]);
        endqstrip();

        n[0] =  -1.;   n[1] =   0.;   n[2] =   0.;   n3f(n);
        bgnqstrip();
        v3f(va[2]);   v3f(va[3]);   v3f(va[6]);   v3f(va[7]);
        endqstrip();
}




defineangle()
{
double theta, dtheta = 2.*M_PI/SIDES;
int i;
        for (i = 0, theta = 0.;  i < SIDES;   i++, theta += dtheta)   {
                angle[i][SIN] = sin(theta);
                angle[i][COS] = cos(theta);
}}



static int die()
{
    fprintf(stderr, "Server closed connection\n");
    exit(1);
}




drawarm(short int dof[DEGREES_OF_FREEDOM], double mats[5][4][4])
{

/* all dimensions are measured in centimeters (cm) */
static double ftr[5][3] = {    /* finger translations [finger #][x,y,z]*/
        {2.6,0.0,0.0},
        {1.0,0.0,0.0},
        {-.8,0.0,0.0},
        {-2.5,0.0,0.0},
        {-1.8,-6.0,0.0}};

static double aang[7] = {
        0,900,450,450,900,300,300};

static double fang[5][4] =
        {{-200,0,-200,-200},
         {-200,0,-200,-200},
         {-200,0,-200,-200},
         {-200,0,-200,-200},
         {-200,0,-200,-200}};

static double delta = {.01};

static double frad[5][4][3] = {
```

```c
/* finger cylinder radii and heights [finger #][digit #][x,y,z] */
        {{1.0,1.8,.8},
         {.95,1.8,.75},
         {.85,1.8,.6},
         {.85,1.5,.55}},          /* pinky */

        {{1.15,2.5,1.15},
         {1.05,2.5,.85},
         {.95,2.0,.75},
         {.9,1.2,.65}},           /* ring finger */

        {{1.1,3.0,1.1},
         {1.1,2.7,1.0},
         {.95,2.2,.8},
        -{.95,1.5,.7}},           /* middle finger */

        {{.85,2.4,.85},
         {.85,2.2,.85},
         {.8,2.0,.75},
         {.75,1.3,.65}},          /* index finger */

        {{1.5,4.0,1.5},
         {1.1,2.5,1.25},
         {1.1,2.0,.9},
         {.85,1.0,.6}}};          /* thumb */

static double sch = { 10.0 };     /* shoulder cap height */

static double arad[4][3] = {
/* arm cylinder radii and heights [digit #][x,y,z] */
        {6.0,30.0,6.0}, /* was 6,30,6 */
        {5.0,25.0,4.5}, /* was 5,25,4.5 */
        {3.0,8.0,3.0},
        {4.0,1.15,1.2}};

static double tang[3] = { 450, -900, 450 };
/* thumb rotation angles x, y, and z */

double x, y, z, dz;
float n[3], v[3];
int i,j, da;
Matrix mvm;  /* modelviewmatrix */

        pushmatrix();
        armang(fang,aang,dof);
/* draw shoulder */
        rotate(aang[1],'x');
        rotate(aang[0],'z');
        drawcap(arad[0][0],sch,arad[0][2]);                 /* cap */
/* draw upper arm */
        rotate(1800,'x');
        translate(0,-.01,0);
        drawcyl(arad[0][0],arad[0][2],arad[1][0],arad[1][2],arad[0][1]);
/* draw elbow */
        translate(0,arad[0][1]-delta,0);
        drawcap(arad[1][0],arad[1][0],arad[1][2]);          /* elbow nuckle */
        rotate(aang[3],'x');
        rotate(aang[2],'z');
/* draw lower arm */
        drawcyl(arad[1][0],arad[1][2],arad[2][0],arad[2][2],arad[1][1]);/* draw
```

```c
        translate(0,arad[1][1]-delta,0);
        drawcap(arad[2][0],arad[2][0],arad[2][2]);  /* wrist nuckle */
        rotate(aang[4],'y');
        rotate(aang[5],'x');
        rotate(aang[6],'z');
/* draw hand */
        drawcyl(arad[2][0],arad[2][2],arad[3][0],arad[3][2],arad[2][1]);
/* cap hand stump */
        translate(0,arad[2][1]-delta,0);
        drawcap(arad[3][0],arad[3][1],arad[3][2]);                    /* cap */

/* draw all 5 fingers */
      for (i=0;i<5;i++) {
        pushmatrix();
        translate(ftr[i][0],ftr[i][1],ftr[i][2]);
        if (i==4) {                  /* thumb */
                rotate(tang[1],'y');
                rotate(tang[0],'x');                 /* remove numbers later */
                rotate(tang[2],'z');
                }
        rotate(fang[i][0],'x');
        rotate(fang[i][1],'z');
drawcyl(frad[i][0][0],frad[i][0][2],frad[i][1][0],frad[i][1][2],frad[i][0][1]);
        translate(0,frad[i][0][1],0);
drawnuckle(frad[i][1][0],frad[i][1][2],-fang[i][2]);     /* nuckle */
        rotate(fang[i][2],'x');
drawcyl(frad[i][1][0],frad[i][1][2],frad[i][2][0],frad[i][2][2],frad[i][1][1]);
        translate(0,frad[i][1][1],0);
drawnuckle(frad[i][2][0],frad[i][2][2],-fang[i][3]);    /*nuckle*/
        rotate(fang[i][3],'x');
drawcyl(frad[i][2][0],frad[i][2][2],frad[i][3][0],frad[i][3][2],frad[i][2][1]);
        translate(0,frad[i][2][1],0);
drawcap(frad[i][3][0],frad[i][3][1],frad[i][3][2]); /* cap */
        translate(0.,frad[i][3][1],0.);
        getmatrix(mvm);
        ftloc[i][0] = mvm[3][0];
        ftloc[i][1] = mvm[3][1];
        ftloc[i][2] = mvm[3][2];
        popmatrix();
      }
        popmatrix();
}

armang(double fng[5][4], double ang[7], short int dof[DEGREES_OF_FREEDOM])

{
int i,j;

        for (i=0;i<7;i++) {
            ang[i] = 10*(double)dof[i];}

        for (i=0;i<5;i++) {
          for (j=0;j<4;j++) {
            fng[i][j] = 10*(double)dof[7+i*4+j];}} /* 10 can be removed */

}

drawcyl(rx1,rz1,rx2,rz2,dy)
double rx1,rz1,rx2,rz2,dy;
{
```

```
double x, y, z;
float n[3], v[3];
int j;

        bgnqstrip();
        for (j = 0;   j <= SIDES;   j++)   {
                if (j == SIDES) {
                        x = angle[0][COS];
                        z = angle[0][SIN];
                }
                else {
                        x = angle[j][COS];
                        z = angle[j][SIN];
                }
                n[0] = x;   n[1] = 0;   n[2] = z;
                n3f(n);
                v[0] = rx1*x;   v[1] = 0.;   v[2] = rz1*z;
                v3f(v);
                v[0] = rx2*x;   v[1] = dy;   v[2] = rz2*z;
                v3f(v);
        }
        endqstrip();
}

drawcap(rx,ry,rz)
double rx,ry,rz;
{
double cosphi, sinphi, cospdp, sinpdp, costheta, sintheta;
double x, y, z;
float n[3], v[3];
int i, j;

        for (i = 0;   i < 4;   i++)   {
                cosphi = angle[i][COS];
                sinphi = angle[i][SIN];
                cospdp = angle[i+1][COS];
                sinpdp = angle[i+1][SIN];
                bgnqstrip();
                for (j = 0;   j <= SIDES;   j++)   {
                        if (j == SIDES) {
                                costheta = angle[0][COS];
                                sintheta = angle[0][SIN];
                        }
                        else {
                                costheta = angle[j][COS];
                                sintheta = angle[j][SIN];
                        }
                        x = costheta * cosphi;
                        y =            sinphi;
                        z = sintheta * cosphi;
                        n[0] = x;   n[1] = y;   n[2] = z;
                        n3f(n);
                        v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                        v3f(v);
                        x = costheta * cospdp;
                        y =            sinpdp;
                        z = sintheta * cospdp;
                        n[0] = x;   n[1] = y;   n[2] = z;
                        n3f(n);
                        v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
```

```
                                        v3f(v);
                        }
                        endqstrip();
                }

                bgntmesh();
                n[0] = 0.;   n[1] = 1.;   n[2] = 0.;
                n3f(n);
                v[0] = 0.;   v[1] = ry;   v[2] = 0.;
                v3f(v);
                for (j = SIDES;  j >= 0;  j--)  {
                        if (j == SIDES) {
                                x = angle[0][COS] * cospdp;
                                y =                 sinpdp;
                                z = angle[0][SIN] * cospdp;
                        }
                        else {
                                x = angle[j][COS] * cospdp;
                                y =                 sinpdp;
                                z = angle[j][SIN] * cospdp;
                        }
                        n[0] = x;   n[1] = y;   n[2] = z;
                        n3f(n);
                        v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                        v3f(v);
                        swaptmesh();
                }
                endtmesh();
}

drawnuckle(rz,rx,anc)
double rx,rz;   /* radius of shpere */
int  anc;   /* angle of next cylinder */
{
double costheta, sintheta, costdt, sintdt, cosphi, sinphi;
double x, y, z;
float n[3], v[3];
int i, j, loop;

        if (anc == 0) return;
        rotate(-900,'y');
        if        (anc <= 180) loop = 1;
        else if (anc <= 360) loop = 2;
        else if (anc <= 540) loop = 3;
        else if (anc <= 720) loop = 4;
        else                 loop = 5;
        for (i = 0;  i < loop;  i++)  {
                costheta = angle[i][COS];
                sintheta = angle[i][SIN];
                costdt   = angle[i+1][COS];
                sintdt   = angle[i+1][SIN];
                bgnqstrip();
                for (j = -5;  j <= 5;  j++)  {
                        if (j < 0) {cosphi = angle[j+SIDES][COS];
                                    sinphi = angle[j+SIDES][SIN];}
                        else {      cosphi = angle[j][COS];
                                    sinphi = angle[j][SIN];}
                        x = cosphi * costheta;
                        y = cosphi * sintheta;
                        z = sinphi;
```

```c
                            n[0] = x;   n[1] = y;   n[2] = z;
                            n3f(n);
                            v[0] = rx*x;   v[1] = rx*y;   v[2] = rz*z;
                            v3f(v);
                            if (i == loop-1) {
                                    x = cosphi * cos(anc*2.*M_PI/3600.);
                                    y = cosphi * sin(anc*2.*M_PI/3600.);
                                    z = sinphi;
                                    n[0] = x;   n[1] = y;   n[2] = z;
                                    n3f(n);
                                    v[0] = rx*x;   v[1] = rx*y;   v[2] = rz*z;
                                    v3f(v);
                            }
                            else {
                                    x = cosphi * costdt;
                                    y = cosphi * sintdt;
                                    z = sinphi;
                                    n[0] = x;   n[1] = y;   n[2] = z;
                                    n3f(n);
                                    v[0] = rx*x;   v[1] = rx*y;   v[2] = rz*z;
                                    v3f(v);
                            }

                    }
                    endqstrip();
            }
    rotate(900,'y');

}
```

```
/*****************************************************/
/*          Jeffrey J. Garside                       */
/*          ADCACS Project                           */
/*          June, 1991                               */
/*          Program:  movbox.c                       */
/*          Features:  Collision Detection,          */
/*                     Rotated Boxes,                */
/*                     and Box Manipulation          */
/*****************************************************/

/* TO COMPILE:  % cc movbox.c -lm -lgl_s -lc_s -s -o movbox <RETURN> */
/* TO EXECUTE:  % movbox iris <RETURN>    OR    % movbox pc2 <RETURN> */

#include <math.h>
#include <gl/gl.h>
#include <gl/device.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <sex.h>

#define SIN   0
#define COS   1
#define _BSD_SIGNALS
#define SIDES    8 /* number of sides to polygons making up arm & fingers */
#define LINE_LEN                80
#define DEGREES_OF_FREEDOM      32
#define SERVER_PORT   5001 /* Server listens for connect requests on port */
#define THUMB         0     /* The order of the fingers */
#define INDEX         1
#define MIDDLE        2
#define RING          3
#define PINKY         4
#define NBR_OF_FNGRS  5      /* This could become 3 and speed up the process */
#define X       0
#define Y       1
#define Z       2
#define XYZ     3

static float incr = .1;/* incremental distance that a box moves when touched */
static int die();
Boolean hit_box[NBR_OF_FNGRS];

Matrix Identity = { 1, 0, 0, 0,
                    0, 1, 0, 0,
                    0, 0, 1, 0,
                    0, 0, 0, 1 };


/* define lighting models */
static float lm[] = {
        AMBIENT, .1, .1, .1,
        LOCALVIEWER, 0,
        LMNULL};

static float lt0[] = {
        LCOLOR, 1., 1., 1.,
```

```
        POSITION, -0.4, 0.4, 1., 0.,   /* was .5, .5, 1., 0. */
        LMNULL};
static float lt1[] = {
        LCOLOR, 1., 1., 1.,
        POSITION, 0.4, -0.4, 1., 0.,   /* was .5, .5, 1., 0. */
        LMNULL};

/* definition of different colored materials */

float mat1[] = {   /* skin */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .999, .400, .240,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat2[] = {   /* blue */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, 0., 1.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat3[] = {   /* green */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, 1., 0.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat4[] = {   /* cyan */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 0., 1., 1.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat5[] = {   /* lt green */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 0., 0.5, 0.1,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat6[] = {   /* yellow */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1., 1., 0.,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat7[] = {   /* pale green */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, .5, 0.05,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat8[] = {   /* orange */
```

```
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1.0, 0.3, 0.0,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};

float mat9[] = {  /* magenta */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1., 0.0, 1.0,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat10[] = {  /* grey */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .4, .4, .4,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat11[] = {  /* lt blue */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, .3, 1.0,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat12[] = {  /* red */
        AMBIENT, .1, .1, .1,
        DIFFUSE, 1.0, 0.0, 0.0,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat13[] = {  /* dk red */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .3, .0, .0,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat14[] = {  /* dk blue */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, .0, .3,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat15[] = {  /* dk purple */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .3, .0, .3,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat16[] = {  /* dk cyan */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .0, .3, .3,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
float mat17[] = {  /* dk grey */
        AMBIENT, .1, .1, .1,
        DIFFUSE, .25, .25, .25,
        SPECULAR, .5, .5, .5,
        SHININESS, 0,
        LMNULL,};
```

```c
/* number of sides on polygons making up the arm, fingers, etc. */
double angle[SIDES][2];

/* storing the location of the fingertips for collision detection */
float ftl[][3] = {
        {0.,0.,0.},
        {0.,0.,0.},
        {0.,0.,0.},
        {0.,0.,0.},
        {0.,0.,0.}};

/* definition of the boxes that appear on the screen */
static float bx[][10] = {

/* { xrad,   yrad,    zrad,
     xang,   yang,    zang,
     xtrans, ytrans, ztrans,
     material number for color} */

{ 5.,  10.,  10.,  300.,    0.,    0.,  -40.,    0.,  10.,   4.},
{10.,  10.,  10.,    0.,  300.,    0.,   -5.,   40.,  10.,   5.},
{10.,  10.,   5.,  700.,    0.,    0.,   40.,    0.,   0.,   2.},
{ 5.,   5.,  10.,    0.,  600.,    0.,    0.,  -40.,   0.,   6.},
{10.,   5.,   5.,    0.,    0.,  450.,    0.,   40.,  10.,   8.},
{ 5.,  10.,   5.,    0.,    0.,  450.,   40.,    0.,  10.,   9.},
{ 5.,   5.,   5.,    0.,    0.,  600.,  -37.,   10.,  10.,  12.},
{10.,   5.,  10.,    0.,    0.,  300.,   10.,  -37.,  10.,  11.},
{ 3.,   3.,   3.,    0.,    0.,    0.,   18.,   18.,   0.,  12.}};

/* how many boxes were defined above */
int NUMBER_OF_BOXES = 9;

/* boolean variable to determine if red/blue box is still being hit */
Boolean prev_hit = FALSE;

/* begin main program */
main(argc,argv)
    int argc;
    char **argv;
{
        int i, cnt, sock;
        struct sockaddr_in sin;
        struct hostent *hp;
        char line[LINE_LEN];
        short dof[DEGREES_OF_FREEDOM];
        long xorigin, yorigin, xsize, ysize;
        float rx, ry;
        short val;

/* zero the arm angles for initial positioning */
    for (i=0;i<32;i++) dof[i] = 0;

/* insure that the program was invoked correctly */
    if (argc != 2) {
        printf("usage: %s host\n", argv[0]);
        exit(1);}

    if ((sock = socket (AF_INET, SOCK_STREAM, 0)) < 0) {
        perror("Can't open socket");
```

```c
        exit(1);}

/* Initialize the socket address to the server's address. */

    bzero((char *) &sin, sizeof(sin));
    sin.sin_family = AF_INET;
    hp = gethostbyname(argv[1]);        /* to get host address */
    if (hp == NULL) {
        herror(argv[1]);
        exit(1);}
    bcopy (hp->h_addr, &(sin.sin_addr.s_addr), hp->h_length);
    sin.sin_port = htons(SERVER_PORT);

/* Connect to the server. */

    if (connect(sock, &sin,sizeof(sin)) < 0) {
        close(sock);
        perror("Connect to server");
        exit(1);}
    printf("Connection established. The data is entered at the server.\n");

/* If the server goes away while sending data, we'll get a        */
/* SIGPIPE signal.  Catch it so we can print an error message.    */

    (void) signal(SIGPIPE, die);

/* size of window */
        prefsize(1280, 1024);

/* define angles once for entire program to speed processor time */
        defineangle();

/* leave the border off and name window */
        noborder();
        winopen("arm");

/* mouse controls and escape functions */
        qdevice(ESCKEY);
        qdevice(LEFTMOUSE);
        qdevice(MIDDLEMOUSE);
        qdevice(RIGHTMOUSE);

/* get xyz values for orientation */
        getorigin(&xorigin, &yorigin);
        getsize(&xsize, &ysize);

/* double buffer to prevent some flicker problems */
        doublebuffer();
        RGBmode();
        gconfig();

/* to increase drawing speed only draw objects with clockwise orientation */
        backface(TRUE);

        lsetdepth(0, 0x7fffff);
        zbuffer(1);
        mmode(MVIEWING);
        loadmatrix(Identity);

/* define how to view the virtual reality */
```

```
        perspective(400, xsize/(float)ysize, 50., 300.0);

/* define all materials for later use, also define lighting models */
        lmdef(DEFMATERIAL, 1, 0, mat1);
        lmdef(DEFMATERIAL, 2, 0, mat2);
        lmdef(DEFMATERIAL, 3, 0, mat3);
        lmdef(DEFMATERIAL, 4, 0, mat4);
        lmdef(DEFMATERIAL, 5, 0, mat5);
        lmdef(DEFMATERIAL, 6, 0, mat6);
        lmdef(DEFMATERIAL, 7, 0, mat7);
        lmdef(DEFMATERIAL, 8, 0, mat8);
        lmdef(DEFMATERIAL, 9, 0, mat9);
        lmdef(DEFMATERIAL,10, 0, mat10);
        lmdef(DEFMATERIAL,11, 0, mat11);
        lmdef(DEFMATERIAL,12, 0, mat12);
        lmdef(DEFMATERIAL,13, 0, mat13);
        lmdef(DEFMATERIAL,14, 0, mat14);
        lmdef(DEFMATERIAL,15, 0, mat15);
        lmdef(DEFMATERIAL,16, 0, mat16);
        lmdef(DEFMATERIAL,17, 0, mat17);
        lmdef(DEFLIGHT, 1, 0, lt0);
        lmdef(DEFLIGHT, 2, 0, lt1);
        lmdef(DEFLMODEL,1, 0, lm);
        lmbind(MATERIAL, 1);
        lmbind(LMODEL, 1);
        lmbind(LIGHT0, 1);
        lmbind(LIGHT1, 2);

/* translate so that viewer's eye is not at origin */
        translate(0, 0, -150);

        while (!(qtest() && qread(&val) == ESCKEY && val == 0)) {

/* Send a synchronizing request for data to the server. */
        strcpy(line, "?");
        cnt = strlen(line);
        if (write(sock, line, cnt) < 0) {
          perror("Error writing to socket.");
          exit(1);}

/* background color */
              czclear(0xffffff,0x7fffff);

/* next objects drawn will have skin color */
              lmbind(MATERIAL,1);

              pushmatrix();

/* draw arm at +60 with rotations as given by mouse position */
              translate(0,0,60.);
              ry = 300.*(2.0*(getvaluator(MOUSEX)-xorigin)/xsize-1.0);
              rx = 300.*(2.0*(getvaluator(MOUSEY)-yorigin)/ysize-1.0);
              rot(ry,'y');
              rot(rx,'x');

/* draw arm */
              drawarm(dof);

              popmatrix();
```

```
/* draw background */
                drawbackground();

                swapbuffers();

/* Read the degrees of freedom array from the socket.*/
            if (read(sock, dof, sizeof(dof)) != sizeof(dof)) {
                printf("Error reading data from server.\n");
                exit(1);}

/* Reverse the byte sex. */
            for (i = 0;   i < DEGREES_OF_FREEDOM;   i++) {
                dof[i] = swap_half(dof[i]);}

}
        printf ("Done, hit CTRL-C to get % prompt if necessary.\n");
        exit(0);
}


drawbackground()
/* This subroutine draws all of the background boxes with their */
/* rotations and translations as defined originally or changed  */
/* by the user's interactions with the boxes on the screen.     */
{
int i;
        for (i = 0;   i < NUMBER_OF_BOXES;   i++) {

                pushmatrix();
/*              translate(bx[i][6],bx[i][7],bx[i][8]); */
                rotate(bx[i][4], 'y');
                rotate(bx[i][3], 'x');
                rotate(bx[i][5], 'z');
                translate(bx[i][6],bx[i][7],bx[i][8]);
                drawbox(i);
                popmatrix();
        }
}

drawbox(ct)
/* this subroutine detects box collision,     */
/* determines which color a box will be,      */
/* moves any boxes which need to be moved,     */
/* and switches the direction of a boxes       */
/* movement if a special box is intersected.  */
int ct;
{

float xr, yr, zr, yht;    /* box radii    */
int fg;                    /* finger index */
int s1,s2,s3,s4,s5,s6;
float v[XYZ],a,b,c;

xr=bx[ct][0];
yr=bx[ct][1];
zr=bx[ct][2];

/* determine which finger is touching the box */
        for (fg=0; fg<NBR_OF_FNGRS; fg++) {
                v[X] = ftl[fg][X];
```

```
            v[Y] = ftl[fg][Y];
            v[Z] = ftl[fg][Z];
            if (eye_to_trans(v)) {
                    printf("Error: 'eye_to_trans'\n");
                    exit(1);
            }
            hit_box[fg] =
                         (v[X] >= -xr) && (v[X] <= xr)
                   && (v[Y] >= -yr) && (v[Y] <= yr)
                   && (v[Z] >= -zr) && (v[Z] <= zr);
      }

if (ct != NUMBER_OF_BOXES - 1) {
/* draw boxes with collision status dependent color */
    if       (hit_box[INDEX] ) colorbox(xr, yr, zr, 13);
    else if  (hit_box[MIDDLE]) colorbox(xr, yr, zr, 14);
    else if  (hit_box[RING]  ) colorbox(xr, yr, zr, 15);
    else if  (hit_box[PINKY] ) colorbox(xr, yr, zr, 16);
    else if  (hit_box[THUMB] ) colorbox(xr, yr, zr, 17);
    else                       colorbox(xr, yr, zr, (int)bx[ct][9]);
```

/* if the boxes are hit, determine which "pyramid" is being hit and move the
box toward the corresponding pyramid's tip. The pyramids are made up of
four corners on the side of any box and the center of the box. The test
for the location of the fingertip is performed by using plane equations
and a cross product function to evaluate which side of the plane the tip
of the finger is located on. */

```
    for(fg=0;fg < NBR_OF_FNGRS ;fg++) {
    if (hit_box[fg]) {
```

/* abc is the xyz location of the fingertip */
```
    a=ftl[fg][0];
    b=ftl[fg][1];
    c=ftl[fg][2];
```

/* test values for plane identification of the points */

```
s1=f(ct,a,b,c,bx[ct][0],bx[ct][1],-bx[ct][2], -bx[ct][0],bx[ct][1],-bx[ct][2]);
s2=f(ct,a,b,c,-bx[ct][0],bx[ct][1],-bx[ct][2],-bx[ct][0],bx[ct][1],bx[ct][2]);
s3=f(ct,a,b,c,-bx[ct][0],bx[ct][1],bx[ct][2],  bx[ct][0],bx[ct][1],bx[ct][2]);
s4=f(ct,a,b,c,bx[ct][0],bx[ct][1],bx[ct][2],   bx[ct][0],bx[ct][1],-bx[ct][2]);
s5=f(ct,a,b,c,bx[ct][0],-bx[ct][1],-bx[ct][2], bx[ct][0],bx[ct][1],-bx[ct][2]);
s6=f(ct,a,b,c,bx[ct][0],bx[ct][1],bx[ct][2],   bx[ct][0],-bx[ct][1],bx[ct][2]);

if (s2 ==  1 && s4 == -1 && s5 ==  1 && s6 ==  1) bx[ct][6] -=  incr;
if (s1 ==  1 && s2 ==  1 && s3 ==  1 && s4 ==  1) bx[ct][7] -=  incr;
if (s1 ==  1 && s3 == -1 && s5 ==  1 && s6 == -1) bx[ct][8] -=  incr;
if (s2 == -1 && s4 ==  1 && s5 == -1 && s6 == -1) bx[ct][6] +=  incr;
if (s1 == -1 && s2 == -1 && s3 == -1 && s4 == -1) bx[ct][7] +=  incr;
if (s1 == -1 && s3 ==  1 && s5 == -1 && s6 ==  1) bx[ct][8] +=  incr;
        }}}
```

/* check special box which determines if a collision pushes or pulls a box */
```
else {if (ct==NUMBER_OF_BOXES-1)              {
        if (hit_box[3] == TRUE  && prev_hit==FALSE)        {
              prev_hit = TRUE;
              incr=incr*-1;                                }
      if (hit_box[3] == FALSE && prev_hit==TRUE )
              prev_hit = FALSE;         }
```

```c
/* color that special box either red or blue */
if(incr>0) colorbox(xr,yr,zr, 2);
else       colorbox(xr,yr,zr,12);


      }

}


int f(j,a,b,c,d,e,f,g,h,i)
/* this function elvaluates the location of the fingertip with respect to  */
/* the particular plane made up of the box corners and the origin.         */
float a,b,c,d,e,f,g,h,i;
int j;
{
/* recall a translation of -150 in the Z
    direction at the beginning of the program */
if(   ((a      -bx[j][6])  *  (e*i-f*h)+
       (b      -bx[j][7])  *  (f*g-d*i)+
       (c+150.-bx[j][8])   *  (d*h-e*g))   >= 0.0)

        return( 1);
   else
        return(-1);
}


int eye_to_trans(v)
/* convert a 3D point from the eye coordinate system */
/* to the transformed coordinate system */
float v[]; /* x, y, z  coordinates of the 3D point */
{
Matrix a;    /* ModelViewMatrix */

float temp;

getmatrix(a);

/* column 0 */
        if (a[0][0] == 0.) {
              if       (a[0][1] != 0.) {
                     temp = a[0][0];  a[0][0] = a[0][1];  a[0][1] = temp;
                     temp = a[1][0];  a[1][0] = a[1][1];  a[1][1] = temp;
                     temp = a[2][0];  a[2][0] = a[2][1];  a[2][1] = temp;
                     temp = a[3][0];  a[3][0] = a[3][1];  a[3][1] = temp;
                     temp =    v[0];     v[0] =    v[1];     v[1] = temp;
              }
              else if (a[0][2] != 0.) {
                     temp = a[0][0];  a[0][0] = a[0][2];  a[0][2] = temp;
                     temp = a[1][0];  a[1][0] = a[1][2];  a[1][2] = temp;
                     temp = a[2][0];  a[2][0] = a[2][2];  a[2][2] = temp;
                     temp = a[3][0];  a[3][0] = a[3][2];  a[3][2] = temp;
                     temp =    v[0];     v[0] =    v[2];     v[2] = temp;
              }
              else return(1);
        }
        a[1][0] /= a[0][0]; a[2][0] /= a[0][0]; a[3][0] /= a[0][0];
            v[0] /= a[0][0];
```

```
        if (a[0][1] != 0.) {
                a[1][1] = a[1][1]/a[0][1] - a[1][0];
                a[2][1] = a[2][1]/a[0][1] - a[2][0];
                a[3][1] = a[3][1]/a[0][1] - a[3][0];
                   v[1] =    v[1]/a[0][1] -    v[0];
        }
        if (a[0][2] != 0.) {
                a[1][2] = a[1][2]/a[0][2] - a[1][0];
                a[2][2] = a[2][2]/a[0][2] - a[2][0];
                a[3][2] = a[3][2]/a[0][2] - a[3][0];
                   v[2] =    v[2]/a[0][2] -    v[0];
        }

/* column 1 */
        if (a[1][1] == 0.) {
                if (a[1][2] == 0.) return(1);
                temp = a[1][1];  a[1][1] = a[1][2];  a[1][2] = temp;
                temp = a[2][1];  a[2][1] = a[2][2];  a[2][2] = temp;
                temp = a[3][1];  a[3][1] = a[3][2];  a[3][2] = temp;
                temp =    v[1];     v[1] =    v[2];     v[2] = temp;
        }
        v[1] /= a[1][1]; a[3][1] /= a[1][1]; a[2][1] /= a[1][1];

        if (a[1][2] != 0.) {
                a[2][2] = a[2][2]/a[1][2] - a[2][1];
                a[3][2] = a[3][2]/a[1][2] - a[3][1];
                   v[2] =    v[2]/a[1][2] -    v[1];
        }

/* column 2 */
        a[3][2] /= a[2][2]; v[2] /= a[2][2];

/* back substitute */
        v[2] = v[2]                                      - a[3][2];
        v[1] = v[1]                    - v[2]*a[2][1] - a[3][1];
        v[0] = v[0] - v[1]*a[1][0] - v[2]*a[2][0] - a[3][0];

        return(0);
}

colorbox(xr, yr, zr, mat_index)
float xr, yr, zr; /* box radii */
int   mat_index;  /* material of box (color) */
{
float n[XYZ], va[8][XYZ];

        lmbind(MATERIAL, mat_index);

        va[0][X] = va[1][X] = va[4][X] = va[5][X] =  xr;
        va[2][X] = va[3][X] = va[6][X] = va[7][X] = -xr;

        va[1][Y] = va[3][Y] = va[5][Y] = va[7][Y] =  yr;
        va[0][Y] = va[2][Y] = va[4][Y] = va[6][Y] = -yr;

        va[0][Z] = va[1][Z] = va[2][Z] = va[3][Z] =  zr;
        va[4][Z] = va[5][Z] = va[6][Z] = va[7][Z] = -zr;


/* define normals and draw the sides of the box */
        n[X] =  0.;  n[Y] =  0.;  n[Z] =  1.;  n3f(n);
```

```
        bgnqstrip();
        v3f(va[0]);  v3f(va[1]);  v3f(va[2]);  v3f(va[3]);
        endqstrip();

        n[X] =  0.;  n[Y] =  1.;  n[Z] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[1]);  v3f(va[5]);  v3f(va[3]);  v3f(va[7]);
        endqstrip();

        n[X] =  0.;  n[Y] =  0.;  n[Z] = -1.;  n3f(n);
        bgnqstrip();
        v3f(va[5]);  v3f(va[4]);  v3f(va[7]);  v3f(va[6]);
        endqstrip();

        n[X] =  0.;  n[Y] = -1.;  n[Z] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[4]);  v3f(va[0]);  v3f(va[6]);  v3f(va[2]);
        endqstrip();

        n[X] =  1.;  n[Y] =  0.;  n[Z] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[4]);  v3f(va[5]);  v3f(va[0]);  v3f(va[1]);
        endqstrip();

        n[X] = -1.;  n[Y] =  0.;  n[Z] =  0.;  n3f(n);
        bgnqstrip();
        v3f(va[2]);  v3f(va[3]);  v3f(va[6]);  v3f(va[7]);
        endqstrip();
}


drawarm(short int dof[DEGREES_OF_FREEDOM])
{

/* all dimensions are measured in centimeters (cm) */
static double ftr[5][3] = {    /* finger translations [finger #][x,y,z]*/
        {2.6,0.0,0.0},
        {1.0,0.0,0.0},
        {-.8,0.0,0.0},
        {-2.5,0.0,0.0},
        {-1.8,-6.0,0.0}};

static double aang[7] = {
        0,1800,0,0,0,0,0};

static double fang[5][4] =
        {{-700,0,-700,-700},
         {-700,0,-700,-700},
         {-700,0,-700,-700},
         {   0,0,   0,   0},
         {-400,0,-400,-400}};

static double delta = {.01};

static double frad[5][4][3] = {
/* finger cylinder radii and heights [finger #][digit #][x,y,z] */
        {{1.0,1.8,.8},
         {.95,1.8,.75},
         {.85,1.8,.6},
         {.85,1.5,.55}},          /* pinky */
```

```c
        {{1.15,2.5,1.15},
         {1.05,2.5,.85},
         {.95,2.0,.75},
         {.9,1.2,.65}},              /* ring finger */

        {{1.1,3.0,1.1},
         {1.1,2.7,1.0},
         {.95,2.2,.8},
         {.95,1.5,.7}},              /* middle finger */

        {{.85,2.4,.85},
         {.85,2.2,.85},
         {.8,2.0,.75},
        -{.75,1.3,.65}},             /* index finger */

        {{1.5,4.0,1.5},
         {1.1,2.5,1.25},
         {1.1,2.0,.9},
         {.85,1.0,.6}}};             /* thumb */

static double sch = { 10.0 };        /* shoulder cap height */

static double arad[4][3] = {
/* arm cylinder radii and heights [digit #][x,y,z] */
        {6.0,30.0,6.0},  /* was 6,30,6 */
        {5.0,25.0,4.5},  /* was 5,25,4.5 */
        {3.0,8.0,3.0},
        {4.0,1.15,1.2}};

static double tang[3] = { 450, -900, 450 };
/* thumb rotation angles x, y, and z */

double x, y, z, dz;
float n[3], v[3];
int i,j, da;
Matrix mvm;   /* modelviewmatrix */

        pushmatrix();
        armang(fang,aang,dof);
/* draw shoulder */
        rotate(aang[1],'x');
        rotate(aang[0],'z');
        drawcap(arad[0][0],sch,arad[0][2]);                 /* cap */
/* draw upper arm */
        rotate(1800,'x');
        translate(0,-.01,0);
        drawcyl(arad[0][0],arad[0][2],arad[1][0],arad[1][2],arad[0][1]);
/* draw elbow */
        translate(0,arad[0][1]-delta,0);
        drawcap(arad[1][0],arad[1][0],arad[1][2]);          /* elbow nuckle */
        rotate(aang[3],'x');
        rotate(aang[2],'z');
/* draw lower arm */
        drawcyl(arad[1][0],arad[1][2],arad[2][0],arad[2][2],arad[1][1]);/* draw
        translate(0,arad[1][1]-delta,0);
        drawcap(arad[2][0],arad[2][0],arad[2][2]);  /* wrist nuckle */
        rotate(aang[4],'y');
        rotate(aang[5],'x');
        rotate(aang[6],'z');
```

```
/* draw hand */
        drawcyl(arad[2][0],arad[2][2],arad[3][0],arad[3][2],arad[2][1]);
/* cap hand stump */
        translate(0,arad[2][1]-delta,0);
        drawcap(arad[3][0],arad[3][1],arad[3][2]);                    /* cap */

/* draw all 5 fingers */
      for (i=0;i<5;i++) {   /* for all fingers (i=0;i<5;i++) */
        pushmatrix();
        translate(ftr[i][0],ftr[i][1],ftr[i][2]);
        if (i==4) {                    /* thumb */
                rotate(tang[1],'y');
                rotate(tang[0],'x');                /* remove numbers later */
                rotate(tang[2],'z');
          _         }
        rotate(fang[i][0],'x');
        rotate(fang[i][1],'z');
drawcyl(frad[i][0][0],frad[i][0][2],frad[i][1][0],frad[i][1][2],frad[i][0][1]);
        translate(0,frad[i][0][1],0);
drawnuckle(frad[i][1][0],frad[i][1][2],-fang[i][2]);    /* nuckle */
        rotate(fang[i][2],'x');
drawcyl(frad[i][1][0],frad[i][1][2],frad[i][2][0],frad[i][2][2],frad[i][1][1]);
        translate(0,frad[i][1][1],0);
drawnuckle(frad[i][2][0],frad[i][2][2],-fang[i][3]);    /*nuckle*/
        rotate(fang[i][3],'x');
drawcyl(frad[i][2][0],frad[i][2][2],frad[i][3][0],frad[i][3][2],frad[i][2][1]);
        translate(0,frad[i][2][1],0);
drawcap(frad[i][3][0],frad[i][3][1],frad[i][3][2]); /* cap */
        translate(0.,frad[i][3][1],0.);
        getmatrix(mvm);
/* define the finger tip locations */
        ftl[i][0] = mvm[3][0];
        ftl[i][1] = mvm[3][1];
        ftl[i][2] = mvm[3][2];
        popmatrix();
    }
        popmatrix();
}


armang(double fng[5][4], double ang[7], short int dof[DEGREES_OF_FREEDOM])
/* get the arm and finger angles from the dof array sent across the network */
{
  int i,j;
  ang[0] = 10*(double)dof[0];
  ang[1] = 10*(double)dof[1];
  if(getbutton(LEFTMOUSE))    ang[2] += 50.;
  if(getbutton(MIDDLEMOUSE))  ang[3] += 50.;
  if(getbutton(RIGHTMOUSE))   ang[5] += 50.;

/*  for (i=0;i<7;i++) ang[i] = 10*(double)dof[i];
  for (i=0;i<5;i++) {
    for (j=0;j<4;j++) {
      fng[i][j] = 10*(double)dof[7+i*4+j];
  } }  */
}


defineangle()
{
/* define some angles for later use since
sin and cos functions eat processor time */
```

```
double theta, dtheta = 2.*M_PI/SIDES;
int i;
for (i = 0, theta = 0.;  i < SIDES;  i++, theta += dtheta)   {
        angle[i][SIN] = sin(theta);
        angle[i][COS] = cos(theta);
    }
}

static int die()
{
    fprintf(stderr, "Server closed connection\n");
    exit(1);
}


drawcyl(rx1,rz1,rx2,rz2,dy)
/* draw a cylinder for the arm and finger representations */
double rx1,rz1,rx2,rz2,dy;
{
double x, y, z;
float n[3], v[3];
int j;

        bgnqstrip();
        for (j = 0;  j <= SIDES;  j++)   {
                if (j == SIDES) {
                        x = angle[0][COS];
                        z = angle[0][SIN];
                }
                else {
                        x = angle[j][COS];
                        z = angle[j][SIN];
                }
                n[0] = x;  n[1] = 0;  n[2] = z;
                n3f(n);
                v[0] = rx1*x;  v[1] = 0.;  v[2] = rz1*z;
                v3f(v);
                v[0] = rx2*x;  v[1] = dy;  v[2] = rz2*z;
                v3f(v);
        }
        endqstrip();
}

drawcap(rx,ry,rz)
/* draw a hemisphere for use as finger tips and shoulder caps */
double rx,ry,rz;
{
double cosphi, sinphi, cospdp, sinpdp, costheta, sintheta;
double x, y, z;
float n[3], v[3];
int i, j;

        for (i = 0;  i < 4;  i++)   {
                cosphi = angle[i][COS];
                sinphi = angle[i][SIN];
                cospdp = angle[i+1][COS];
                sinpdp = angle[i+1][SIN];
                bgnqstrip();
                for (j = 0;  j <= SIDES;  j++)   {
                        if (j == SIDES) {
```

```c
                                        costheta = angle[0][COS];
                                        sintheta = angle[0][SIN];
                }
                else {
                                        costheta = angle[j][COS];
                                        sintheta = angle[j][SIN];
                }
                x = costheta * cosphi;
                y =             sinphi;
                z = sintheta * cosphi;
                n[0] = x;   n[1] = y;   n[2] = z;
                n3f(n);
                v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                v3f(v);
                x = costheta * cospdp;
                y =             sinpdp;
                z = sintheta * cospdp;
                n[0] = x;   n[1] = y;   n[2] = z;
                n3f(n);
                v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                v3f(v);
            }
            endqstrip();
        }

        bgntmesh();
        n[0] = 0.;   n[1] = 1.;   n[2] = 0.;
        n3f(n);
        v[0] = 0.;   v[1] = ry;   v[2] = 0.;
        v3f(v);
        for (j = SIDES;   j >= 0;   j--)   {
                if (j == SIDES) {
                        x = angle[0][COS] * cospdp;
                        y =                 sinpdp;
                        z = angle[0][SIN] * cospdp;
                }
                else {
                        x = angle[j][COS] * cospdp;
                        y =                 sinpdp;
                        z = angle[j][SIN] * cospdp;
                }
                n[0] = x;   n[1] = y;   n[2] = z;
                n3f(n);
                v[0] = rx*x;   v[1] = ry*y;   v[2] = rz*z;
                v3f(v);
                swaptmesh();
        }
        endtmesh();
}

drawnuckle(rz,rx,anc)
/* draw a knuckle so hemispheres won't have to be used for knuckles */
double rx,rz;   /* radius of shpere */
int  anc;   /* angle of next cylinder */

double costheta, sintheta, costdt, sintdt, cosphi, sinphi;
double x, y, z;
float n[3], v[3];
int i, j, loop;
```